

# UNIVERSIDAD DE CUENCA



## Facultad de Ingeniería

## Escuela de Informática

### **“Proceso de anotación de Objetos de Aprendizaje semiautomático mediante tecnologías semánticas”**

Trabajo de Titulación previo a la obtención  
del Título de Ingeniero en Sistemas

#### **AUTORES:**

José Hernán Farfán Fajardo

C.I. 0105406102

Juan Carlos Lojano Uguña

C.I. 0105056196

#### **DIRECTOR:**

Ing. Jorge Luis Bermeo Conto, Msc.

C.I. 0103900387

#### **CO-DIRECTOR:**

Ing. Víctor Hugo Saquicela Galarza, PhD.

C.I. 0103599577

Cuenca - Ecuador

2017

## Resumen

Actualmente internet dispone de una inmensa cantidad de Objetos de Aprendizaje (OA) almacenados y disponibles en repositorios (ROA). Estos OA contienen metadatos que describen sus características y están definidos bajo algún estándar, como Dublin Core. Estos brindan los beneficios de la reutilización sin tener que invertir grandes costos, y es ahí donde radica su importancia, especialmente dentro del campo de la educación actual, la cual tiene una tendencia mayormente virtual. Por otro lado, la web semántica se ha convertido en una de las actividades más recientes en los desarrollos orientados a la web. Utiliza ontologías para definir de manera formal conceptos y relaciones entre entidades o clases. Además, existen ontologías específicas para el manejo de OA, como por ejemplo LOM2OWL. Sin embargo, el problema principal es que los OA al no estar descritos en una ontología, son netamente sintácticos, lo que limita su utilización dentro de un campo semántico, y para que los OA estén descritos semánticamente deben estar descritos y poblados en una ontología. Este proceso de poblado consiste en extraer y representar, de manera manual, semiautomática o automática los individuos, las clases, las relaciones y los valores de las propiedades. El objetivo del presente proyecto es el desarrollo de un proceso de anotación de OA mediante tecnologías semánticas, implementando un proceso semiautomático, que reduce el número de pasos y por ende el tiempo de dicho proceso. Los resultados y hallazgos se presentan en el estudio de caso correspondiente, donde se presentan los procesos de poblado manual y semiautomático.

**Palabras Clave:** OBJETOS DE APRENDIZAJE, REPOSITARIOS, METADATOS, ONTOLOGIAS, LOM, DUBLIN CORE, PROTOCOLO OAI-PMH, ANOTACIONES SEMANTICAS, POBLADO DE ONTOLOGIAS, DBPEDIA, DSPACE.

## Abstract

At the present time, internet has a huge amount of learning objects (LO), which are available and stored in repositories (LOR). These LO include metadata which describe their characteristics and they are defined by a kind of standard like Dublin Core. They provide the benefits of reuse without the need to invest a lot of money. And this is the root of the importance, especially in the current educational system which has mostly an online trend. On the other hand, the semantic web has become in one of the most recent activities oriented to the web development. It uses ontologies to define concepts and relationships between entities or types in a formal way. Furthermore, there are specific ontologies for the LO management, such as LOM2OWL. However, the main problem is that LO are not described in an ontology, they are purely syntactic which limit their use in the semantic field. In order, LO can be described semantically, they must be described and populated in an ontology. This populated process involves to take and represent manually, semi-automatic or automatic individuals, types, relationships, and properties values. The objective of this project is the development of LO entry process through semantic technology, implementing a semi-automatic process. The results and findings are presented in the relevant case study, and where the manual and semi-automatic populated process are showed.

**Key Words:** LEARNING OBJECTS, REPOSITORIES, METADATA, ONTOLOGIES, LOM, DUBLIN CORE, OAI-PMH PROTOCOL, SEMANTICS NOTES, ONTOLOGIES POPULATED, DBPEDIA, DSPACE.



# Contenido

## Contenido

Resumen .....	2
Abstract .....	3
Contenido .....	4
Índice de Tablas .....	7
Índice de Imágenes .....	8
Agradecimientos.....	14
Agradecimiento .....	15
Dedicatoria .....	15
Agradecimiento .....	16
Dedicatoria .....	16
Capítulo 1.    Introducción.....	17
1.1    Panorama General.....	17
1.2    Antecedentes .....	17
1.3    Planteamiento del problema y justificación .....	19
1.4    Problemática .....	20
1.5    Justificación.....	21
1.6    Objetivos .....	21
1.6.1    Objetivo general.....	21
1.6.2    Objetivos específicos .....	21
1.7    Metodología de la investigación .....	22
1.8    Estructura del proyecto .....	23
Capítulo 2.    Marco Teórico .....	25
2.1    Introducción.....	25
2.2    Objetos de Aprendizaje .....	25
2.3    Repositorios de Objetos de Aprendizaje (ROA) .....	26
2.3.1    Ágora (Asistencia para la Gestión de Objetos Reusables de Aprendizaje.).....	27
2.3.2    MERLOT (Multimedia Educational Resource for Learning and Online Teaching).....	28



2.3.3	Plataforma ARIADNE.....	29
2.3.4	Tabla comparativa de ROA.....	30
2.4	Metadatos .....	31
2.4.1	Estándares de metadatos .....	31
2.4.1.1	LOM.....	31
2.4.1.2	Dublin Core.....	35
2.4.1.3	SCORM (Sharable Content Object Reference Mode) .....	38
2.4.1.4	IMS (Instruction Management Systems).....	39
2.5	Extracción de metadatos mediante servicios web.....	39
2.5.1	Protocolos para la extracción de metadatos de los repositorios. ....	40
2.5.1.1	Protocolo Z39.50 .....	40
2.5.1.2	Protocolo OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting).....	41
2.6	Objetos de Aprendizaje dentro del marco de la Web Semántica .....	45
2.6.1	Anotaciones Semánticas.....	46
2.6.1.1	Lenguaje OWL (Ontology Web Language).....	46
2.6.2	Ontologías basadas en estándares de metadatos.....	47
2.6.2.1	LOM2OWL (estándar LOM).....	47
2.6.2.2	PROV-FOAF .....	50
2.6.2.3	LOCO (Learning Object Context Ontologies) .....	52
2.6.2.4	Ontología del dominio del Sistema Generador de Ambientes de Enseñanza - Aprendizaje Constructivistas Basados en Objetos de Aprendizaje (AMBAR) 53	
2.7	Población de Ontologías .....	53
2.7.1	Poblado Manual .....	55
2.7.2	Poblado Semiautomático .....	56
2.7.3	Poblado Automático .....	56
2.8	Herramientas Usadas.....	57
2.9	Conclusiones del capítulo.....	63
Capítulo 3.	Diseño de la solución .....	65
3.1	Introducción.....	65
3.2	Descripción de la solución propuesta.....	65

3.3	Entorno de trabajo de la aplicación .....	66
3.4	Repositorios de OA .....	68
3.5	Extracción de metadatos a través del protocolo OAI-PMH .....	69
3.6	Ontologías.....	72
3.7	Mapeo manual y generación de reglas de mapeo .....	73
3.7.1	Identificación de reglas de mapeo .....	74
3.7.2	Representación XML de las reglas de mapeo .....	75
3.8	Proceso de poblado de ontologías.....	76
3.8.1	Proceso semiautomático de poblado de ontologías.....	76
3.8.2	Ontología poblada.....	79
3.9	Vinculación de OA con recursos de DBpedia.....	79
3.10	Almacenamiento del archivo RDF resultante en Apache Marmotta .....	80
3.11	Recapitulación .....	81
Capítulo 4.	Estudio de Caso. Experimento controlado Población de Ontologías	82
4.1	Introducción.....	82
4.2	Estudio de caso.....	82
4.2.1	Selección y definición del caso .....	82
4.2.2	Elaboración de una lista de preguntas .....	85
4.2.3	Localización de las fuentes de datos .....	86
4.2.4	Análisis e interpretación .....	86
4.2.5	Elaboración de un informe .....	86
4.3	Análisis de resultados .....	96
4.3.1	Respuestas a las preguntas del estudio de caso .....	96
4.3.2	Comparación entre los procesos de poblado: manual y semiautomático	99
4.3.3	Vinculación con fuentes externas .....	100
4.4	Interpretación de resultados y conclusiones del experimento .....	106
4.5	Recapitulación.....	107
Capítulo 5.	Conclusiones y trabajos futuros .....	108
1.1	Conclusiones.....	108
1.2	Líneas de trabajo futuras.....	109

Glosario .....	110
Anexos .....	111
Instalación y configuración de Dspace .....	111
Prerrequisitos de software .....	111
Instalación de software necesario .....	111
Instalación de Dspace .....	112
Creación del repositorio local de OA en Dspace .....	114
Subir los OA al repositorio de OA .....	115
Proceso manual de poblado de una ontología .....	120
Referencias .....	131

## Índice de Tablas

Tabla 2.1 Comparación de los ROA. Fuente: autores .....	31
Tabla 2.2 Elementos de tipo educativo en LOM .....	34
Tabla 2.3 Grupos principales del estándar Dublin Core. Fuente: autores .....	35
Tabla 2.4 Elementos de Dublin Core. Fuente: <a href="https://goo.gl/pr7nKM">https://goo.gl/pr7nKM</a> .....	38
Tabla 2.5 Principales clases ontológicas con sus respectivas categorías y subcategorías LOM. ....	50
Tabla 4.2 Resultados del proceso de poblado semiautomático .....	95
Tabla 4.3 Resultados de población manual y población semiautomática .....	99
Tabla 4.4 Enlaces DBpedia generados para el OA: El Lenguaje HTML que contiene un solo Subject .....	100
Tabla 4.5 Enlaces DBpedia generados para un OA titulado Relación entre Apuntadores y Cadenas, que contiene varios Subjects .....	100
Tabla 4.6 OA titulado Do_While_Ejercicio no genera enlaces de DBpedia .....	101
Tabla 4.7 Criterios de calificación manual .....	101
Tabla 4.8 Relación entre los OA y los enlaces generados. ....	102
Tabla 4.9 Relación entre OA y enlaces DBpedia, Subjects y links .....	103
Tabla 4.10 Relación entre los OA y los enlaces, con promedios devueltos por FinalScore .....	104
Tabla 4.11 Promedios de FinalScore y promedios de calificación manual .....	105
Tabla 1 Descripción del OA a poblar en la ontología .....	123

# Índice de Imágenes

Imagen. 2.1 Repositorio Ágora. Búsqueda de OA referentes a programación...	28
Imagen. 2.2 Repositorio Merlot. Búsqueda de recursos referentes a programación.	29
Imagen. 2.3 Interfaz de Ariadne Finder .....	29
Imagen. 2.4 Categorías LOM. Fuente: <a href="https://goo.gl/fRdaf5">https://goo.gl/fRdaf5</a> .....	32
Imagen. 2.5 Elementos de categoría General.....	34
Imagen. 2.6 Peticiones en el protocolo OAI-PMH. Fuente: <a href="https://goo.gl/6RWNup">https://goo.gl/6RWNup</a> .....	42
Imagen. 2.7 Llamada a servicio web usando el protocolo OAI-PMH. Fuente: autores .....	43
Imagen. 2.8 Extracción de colecciones o grupos de recursos. Fuente: autores	43
Imagen. 2.9 Servicio web que permite la extracción de metadatos del Library of Congress Open Archive Initiative Repository. Fuente: autores .....	43
Imagen. 2.10 Servicio web que permite la extracción de formatos de metadatos. Fuente: autores .....	44
Imagen. 2.11 Servicio web que permite la extracción de un conjunto de metadatos referentes a la música. Fuente: autores .....	44
Imagen. 2.12 Servicio web que permite la extracción de un conjunto de identificadores por el prefijo oai_dc. Fuente: autores .....	44
Imagen. 2.13 Servicio web que permite la extracción de los metadatos de un O, por el prefijo oai_dc y por un identificador. Fuente: autores .....	44
Imagen. 2.14 Resultado de la consulta de la imagen 2.20. Extracción de metadatos de un OA. Fuente: autores .....	45
Imagen. 2.15 Anotación Semántica a una Ontología. Fuente Semantic Annotation, Indexing, and Retrieval.....	46
Imagen. 2.16 Relaciones directas de la clase learningObject del estándar LOM. Fuente: Autores.....	48
Imagen. 2.17 Entidad, Actividad y Agente (PROV). Fuente <a href="http://ceweb.br/guias/web-semantica/es/capitulo-6/">http://ceweb.br/guias/web-semantica/es/capitulo-6/</a> .....	51
Imagen. 2.18 Jerarquía de clases en LOM2OWL. Fuente: <a href="http://www.web.upsa.es/spdece08/contribuciones/176_Fermoso_Sanchez_Sicilia_LOMOWL.pdf">http://www.web.upsa.es/spdece08/contribuciones/176_Fermoso_Sanchez_Sicilia_LOMOWL.pdf</a> .....	54
Imagen. 2.19 Interfaz de Protege para el proceso de poblado manual de una ontología .....	56
Imagen. 2.20 Dspace de la ESPOL (Escuela Superior Politécnica del Litoral). Búsqueda de recursos referentes a programación.....	59
Imagen. 2.21 Comunidad, Colección, ítem de Dspace .....	59
Imagen. 2.22 Ejemplo 1. Listado de películas en español .....	62



Imagen. 2.23 Resultado de consulta SPARQL del ejemplo 1 .....	62
Imagen. 2.24 Ejemplo 2. Director de una película.....	63
Imagen. 2.25 Resultado de consulta SPARQL del ejemplo 2 .....	63
Imagen. 3.1 Descripción de la solución propuesta. Fuente: Autores.....	66
Imagen. 3.2 Menú principal Metadata Harvester. Fuente: Autores .....	67
Imagen. 3.3 Interfaz para el mantenimiento de información de repositorios. Fuente: Autores.....	67
Imagen. 3.4 Interfaz para cargar archivos de configuración. Fuente: Autores. ....	67
Imagen. 3.5 Interfaz para el proceso de poblado de ontologías. Fuente: Autores. .....	68
Imagen. 3.15 Ontología resultante almacenada en un servidor Apache Marmotta. Fuente: Autores.....	80
Imagen. 4.1 Búsqueda de OA en el repositorio Agora, por el criterio de “programación”. Cantidad total de OA obtenidos del ROA Ágora (46) .....	83
Imagen. 4.2. Escenario en el cual es ejecutado el experimento controlado .....	86
Imagen. 4.3 Módulo de Repositorios - Dspace.....	87
Imagen. 4.4 Selección del repositorio Dspace en la aplicación.....	87
Imagen. 4.5 Módulo de extracción de metadatos.....	88
Imagen. 4.6 Elección de grupo de OA: Repositorio de Objetos de Aprendizaje (localizado en el pc1) .....	88
Imagen. 4.7 Selección de todos los OA del repositorio. ....	89
Imagen. 4.8 Módulo de Ontologías .....	89
Imagen. 4.9 Cargado de archivos en la aplicación “Metadata Harvester” .....	90
Imagen. 4.10 Módulo Reglas de mapeo.....	90
Imagen. 4.11 Identificación grafica de reglas de mapeo. Fuente: Autores.....	91
Imagen. 4.12 Interfaz de la aplicación que implementa la solución. Pantalla de bienvenida.....	92
Imagen. 4.13 Módulos Componente desarrollado y ontología poblada.....	93
Imagen. 4.14 Módulo: Vinculación de OA con recursos de DBpedia.....	94
Imagen. 4.15 Módulo Almacenamiento en Apache Marmotta y consultas SPARQL.....	94
Imagen. 4.16 Consulta SPARQL sobre la ontología poblada.....	95
Imagen. 4.17 Visualización en Protege, de la ontología poblada con los metadatos .....	96
Imagen. 4.18 Tiempos de poblado por OA (tiempo en milisegundos) .....	98
Imagen. 4.19 Gráfica de la relación entre OA y enlaces DBpedia.....	102
Imagen. 4.20 Gráfica que describe la relación entre Subjects y links de OA ....	103
Imagen. 4.21 Gráfica de la relación entre los OA y los enlaces, con promedios devueltos por FinalScore.....	104
Imagen. 4.22 Promedios comparativos entre calificación manual y estadística de DBpedia .....	105



Universidad de Cuenca  
Clausula de propiedad intelectual

*José Hernán Farfán Fajardo*, autor del Trabajo de Titulación “Proceso de anotación de Objetos de Aprendizaje semiautomático mediante tecnologías semánticas”, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 10 de Mayo del 2017



José Hernán Farfán Fajardo

C.I: 0105406102



Universidad de Cuenca  
Clausula de derechos de autor

*José Hernán Farfán Fajardo*, autor del Trabajo de Titulación “Proceso de anotación de Objetos de Aprendizaje semiautomático mediante tecnologías semánticas”, reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de Ingeniero en Sistemas. El uso que la Universidad de Cuenca hiciera de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor

Cuenca, 10 de Mayo del 2017



José Hernán Farfán Fajardo

C.I: 0105406102



Universidad de Cuenca  
Clausula de propiedad intelectual

*Juan Carlos Lojano Uguña*, autor del Trabajo de Titulación “Proceso de anotación de Objetos de Aprendizaje semiautomático mediante tecnologías semánticas”, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 10 de Mayo del 2017

Juan Carlos Lojano Uguña

C.I: 0105056196



Universidad de Cuenca  
Clausula de derechos de autor

*Juan Carlos Lojano Uguña*, autor del Trabajo de Titulación "Proceso de anotación de Objetos de Aprendizaje semiautomático mediante tecnologías semánticas", reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de Ingeniero en Sistemas. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor

Cuenca, 10 de Mayo del 2017



Juan Carlos Lojano Uguña

C.I: 0105056196

## Agradecimientos

El presente trabajo de titulación no hubiera sido posible sin la ayuda de nuestro director de tesis, Ing. Jorge Bermeo y nuestro co-director, Ing. Víctor Saquicela, a quienes hacemos un especial agradecimiento por su apoyo y ayuda incondicional a lo largo del desarrollo del presente trabajo. Gracias ingenieros por la confianza y el apoyo brindado durante todo el proceso.

Agradecemos también a todos los docentes que han sido parte de nuestra carrera universitaria, ya que cada uno de ellos han sido una pieza importante para culminar con éxito esta meta.

Finalmente agradecemos a todas las personas que de una u otra forma nos apoyaron en la realización y culminación del presente trabajo de titulación, así como también a lo largo de nuestra carrera universitaria.

José Farfán

Juan Carlos Lojano

## Agradecimiento

En primer lugar, le agradezco a mi Señor Jesús, porque todo lo que soy se lo debo a Él, mi casa, mi familia, mis amigos, mis metas y mis sueños, y por supuesto este trabajo de titulación también se lo debo a Él. Gracias Señor por ser mi Dios y mi Señor. Te agradezco por ayudarme a terminar esta meta. También quiero agradecerle a toda mi familia, desde el más pequeño hasta el más grande, porque cada uno de ellos han sido un apoyo incondicional a lo largo de toda mi carrera. Gracias también a cada uno de mis profesores y mis compañeros y amigos y por supuesto a mi director y codirector del presente trabajo de titulación

Juan Carlos

## Dedicatoria

Primeramente, quiero dedicar este trabajo de titulación a mi Señor Jesús, ya que gracias a Él he podido lograr esta meta y con su ayuda he salido delante de muchas pruebas difíciles de mi vida. En segundo lugar, le dedico este trabajo a toda mi familia, porque cada uno de ellos han sido parte de mis sueños y mis logros. Y también de forma muy especial le dedico este trabajo a cada una de las muchas personas especiales que Dios ha puesto en mi camino, que de una u otra forma han sido parte especial de mi vida.

Juan Carlos

## Agradecimiento

En primer lugar, agradezco a mi familia, empezando por mis padres, quienes me apoyan y me motivan a ser mejor cada día. A mis compañeros, con quienes hemos vivido tantos buenos momentos y formado estrechos lazos de amistad. Finalmente, a mis profesores, por compartir su conocimiento y motivarnos a seguir adelante siempre.

José

## Dedicatoria

Dedico este trabajo a mi principal fuente de motivación, mi familia, que siempre me brindó su apoyo oportuno e incondicional. La mejor recompensa de cada meta alcanzada es disfrutar con ellos los frutos obtenidos.

José



# Capítulo 1. Introducción

## 1.1 Panorama General

En la actualidad existe una gran cantidad de objetos de aprendizaje (OA) disponibles en Internet. Un OA es un conjunto de recursos, auto contenible, diseñado y creado en pequeñas unidades digitales, con un propósito educativo y con el fin de maximizar el número de situaciones en las que se puede utilizar (Hernández Laz & Alonso Reyes, 2014). Entonces, los OA disponen de contenidos reutilizables sin tener que invertir grandes costos, es ahí donde radica la importancia de estos recursos dentro del campo de la educación actual, ya que hoy en día esta tiene una tendencia mayormente virtual. Un OA contiene metadatos, que permiten describir sus características como su título (title), descripción (description), entre otros. Estos generalmente están definidos bajo un estándar, entre los más importantes están: LOM y Dublin Core y tienen la finalidad de facilitar la búsqueda actual de OA ubicados en repositorios como Ágora, Merlot, etc.

Por otro lado, una de las actividades recientes en los desarrollos orientados a la web, es la web semántica, la cual utiliza ontologías para definir de manera formal, conceptos y relaciones entre los tipos, las propiedades, y las relaciones existentes entre entidades o clases (Hernández Laz & Alonso Reyes, 2014). Es importante señalar que existen ontologías, como LOM2OWL, PROV, FOAF, que son específicas para el manejo de OA, lo que permitirá describirlos semánticamente.

De este modo, las tendencias de desarrollo de la web semántica se centran en tres áreas aplicadas a la educación: la informática, el diseño instruccional y los sistemas de bibliotecas. Actuando como pegamento entre estas tres áreas se encuentran los OA. (Santacruz-Valencia, Aedo, & Delgado Kloos, 2004)

## 1.2 Antecedentes

A continuación, se presenta un breve análisis del estado del arte actual con respecto a los trabajos relacionados al manejo de ontologías de OA dentro de la web semántica y todo lo que esto involucra (extracción de metadatos, poblado de ontologías, etc.)

En el trabajo titulado “Learning Object Indexing Tool Based on a LOM Ontology”, sus autores describen un enfoque para indexar OA, esto como una solución para la

gestión y acceso de estos recursos. Además, presentan la construcción de una ontología y también la herramienta que han desarrollado para indexar y recuperar OA, la misma que está basada en una ontología LOM (Ouafia Ghebghoub, Marie-Hélène Abel, & Claude Moulin, 2008). Sin embargo, no se trabaja directamente sobre el poblado de ontologías.

En el trabajo titulado “Una ontología en OWL para la representación semántica de objetos de aprendizaje” sus autores definen una ontología para el modelo de descripción de objetos de aprendizaje IEEE LOM, la misma que es denominada como LOM2OWL y está implementada en el lenguaje OWL usando la herramienta Protege (Ana Mª Fermoso García, Salvador Sánchez Alonso, & Miguel A. Sicilia, 2010). Sin embargo, este trabajo únicamente define la ontología mencionada, mas no la población de la misma.

En el trabajo denominado “Desarrollo de una ontología para la conceptualización de un ambiente virtual de aprendizaje constructivista” sus autores plantean el desarrollo de una ontología llamada AMBAR (Sistema Generador de Ambientes de Enseñanza-Aprendizaje Constructivistas basados en Objetos de Aprendizaje), basada en los estándares SCORM (Sharable Content Object Reference Model) y la especificación IMS Learning Design (IMS LD)(«Desarrollo de una ontología para la conceptualización de un ambiente virtual de aprendizaje constructivista», 2008). Sin embargo, este trabajo únicamente desarrolla la ontología mencionada, mas no la población de la misma.

En la tesis denominada “Planificación TFC / Diseño y población semiautomática de ontologías.” Su autor plantea el desarrollo de una aplicación que permita la instanciación de una ontología previamente creada, así como su navegación. En esta tesis el autor utiliza la herramienta de Protege para crear paso a paso una ontología relacionada con el futbol y además presenta una pequeña aplicación para la población semiautomática de dicha ontología mediante una consulta SPARQL. (Daniel Fiel Cortes, 2016). Sin embargo, el trabajo se enfoca en la población semiautomática mediante consultas SPARQL internas, que devuelven resultados individuales, en este caso de jugadores de futbol.

En el trabajo de tesis denominado “OAEARCH: Modelo de aplicación basado en web semántica para la búsqueda de objetos de aprendizaje mediante perfilado de consultas.” Su autor, presenta un nuevo enfoque para la construcción de motores de búsqueda para objetos de aprendizaje basado en la Web Semántica, así mismo menciona que es posible construir conceptos de búsqueda que son especificados

en una ontología de dominio. Además, plantea el desarrollo de un prototipo de aplicación informática (motor de búsqueda) que demuestre la factibilidad de su enfoque (Gómez Álvarez, 2014). Sin embargo, trabaja únicamente bajo el estándar LOM para OA.

En base a estos antecedentes, y al panorama general presentado, se plantea las siguientes preguntas de investigación:

- ¿Cuál es el proceso de poblado de una ontología?
- ¿Cuántas maneras existen de poblar una ontología? ¿y cuáles son?
- ¿Cuántos estándares existen para metadatos de OA? ¿Es posible poblar ontologías usando cualquier estándar?
- Al agilizarse el proceso de poblado de ontologías ¿Cuánto tiempo se reduciría el mismo?
- ¿Cuántos OA se pueden poblar simultáneamente?

### 1.3 Planteamiento del problema y justificación

Actualmente, la mayoría de buscadores web realizan su labor mediante palabras clave (keywords), estadísticas de popularidad, arboles jerárquicos por temas, etc. (Alejandro Alós Moya, 2011). Su funcionamiento de manera general se basa en el número de veces en la que la palabra es citada en todo el documento, la palabra que se repita el mayor número de veces terminará siendo el descriptor de ese documento. Normalmente, esto se hace en todos los ámbitos de investigación, incluido los OA.

Básicamente, cuando la búsqueda de un OA da como resultado un listado de direcciones Web en donde se mencionan temas relacionados con los keywords del OA que se introdujeron en el buscador, esto produce resultados inexactos o demasiado extensos, en donde lo siguiente a realizar es una búsqueda de forma manual, dando como consecuencia una mayor dificultad para obtener la información deseada y especialmente un gasto exagerado de tiempo en la búsqueda de un OA. En otras palabras, los problemas que presenta este tipo de búsqueda son (Santiago Fernando Suárez Sánchez, 2010):

- Escasa precisión en los resultados
- Alta sensibilidad al vocabulario empleado en la búsqueda
- No incorpora mecanismos para el procesamiento automático de la información

- No incluye mecanismo para la interoperabilidad de los sistemas de información

Sin embargo, si toda la web tuviera la capacidad de construir una base de conocimiento sobre las preferencias y necesidades del usuario, sería posible obtener resultados más exactos y útiles de cualquier búsqueda, sin tener que invertir demasiado tiempo; es así que surge el concepto de la web semántica. De acuerdo con el consorcio W3C, que es la World Wide Web, la cual es un consorcio internacional que genera recomendaciones y estándares que aseguran el crecimiento de la W3C, “La Web Semántica es una Web extendida, dotada de mayor significado en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla”. Dentro de este marco de estudio, un buscador semántico es una herramienta que permite realizar en la Web mediante el uso de algoritmos que simulan comprensión o entendimiento. El problema más significativo respecto a la web semántica es que su implementación es compleja, razón por la cual aún no se cuenta con una web totalmente semántica. Por ende, los beneficios que ofrece una búsqueda semántica respecto a una sintáctica son: resultados concretos y disminución del tiempo de búsqueda (Santiago Fernando Suárez Sánchez, 2010).

## 1.4 Problemática

Un buscador actual produce resultados inexactos o demasiado extensos, que obliga a realizar un proceso de filtrado manual, dando como consecuencia una mayor dificultad para obtener la información deseada y especialmente un gasto exagerado de tiempo en la búsqueda de un OA.

Asimismo, las ontologías son elementos importantes de la web semántica y por ende también de los buscadores semánticos. Dentro de las ontologías usadas para describir OA podemos mencionar: LOM2OWL, LOCO, etc.; las mismas que están desarrolladas para trabajar con los metadatos de OA y bajo alguno de sus estándares (LOM, Dublin Core, etc.). El proceso de poblado de una ontología consiste en extraer y representar, de manera automática o semiautomática los individuos de clase, relaciones entre los individuos y valores de las propiedades (Maricela Bravo José A. Reyes-Ortiz, Oscar Herrera, & Alejandro Gudiño, 2015). Es posible realizar este proceso de poblado de forma manual, a través de herramientas como Protege; pero de esta manera se vuelve una tarea larga y tediosa.

Por lo tanto, la problemática que se identifica, básicamente es que, al realizar consultas con los buscadores actuales, los mismos devuelven resultados inexactos y demasiado extensos. Además, entendiendo que es necesario tener ontologías

pobladas, una problemática importante que se identifica, es el tiempo excesivo que toma poblar de forma manual las ontologías.

## 1.5 Justificación

Un buscador semántico de OA permitiría obtener resultados exactos y útiles, sin tener que invertir demasiado tiempo en ello. Una situación similar ocurre con el proceso de poblado de una ontología, ya que como se mencionó, esto toma demasiado tiempo al realizarlo de forma manual, pero al contar con un proceso que agilite esto, se lograría tener ontologías pobladas con una reducción considerable del tiempo invertido en esta tarea. Y al tener ontologías pobladas las mismas se vuelven útiles para la web semántica, en este caso aplicable a los OA. Es en este punto donde surge la necesidad de desarrollar un sistema que facilite dicho proceso de poblado de las ontologías de OA, y que dé como resultado un archivo RDF poblado con metadatos de OA que pueda ser explotado mediante consultas SPARQL, de tal forma que su conocimiento pueda ser aprovechado.

En conclusión, el presente proyecto plantea el desarrollo de un sistema que agilite el proceso de poblado de OA usando tecnologías semánticas, a través de ontologías y metadatos de OA, extrayendo los mismos de un repositorio, bajo el protocolo OAI-PMH, y bajo estándares como LOM o DUBLIN CORE para OA. Esto mediante un archivo de configuración XML que permitirá cargar diferentes ontologías, con varios metadatos y OA, obteniendo finalmente un archivo RDF que contendrá a los OA poblados en la ontología, permitiendo hacer consultas sobre estos y relacionar estos resultados con fuentes externas, como por ejemplo DBpedia. Y la hipótesis que se plantea es que esto reducirá considerablemente el número de pasos y por ende el tiempo de poblado de ontologías de OA. Además, es importante señalar que todo se podrá usar como base para el desarrollo de un buscador semántico de OA.

## 1.6 Objetivos

### 1.6.1 Objetivo general

Definir un proceso de anotación de OA usando tecnologías semánticas para reducir el tiempo de poblado de ontologías.

### 1.6.2 Objetivos específicos

1. Analizar los OA dentro del marco de la Web Semántica.
2. Analizar el proceso manual, semiautomático y automático de población de una ontología.

3. Desarrollar la aplicación para la anotación semántica.
4. Poblar las ontologías mediante la aplicación y analizar los resultados obtenidos.

## 1.7 Metodología de la investigación

La metodología que se plantea para el presente proyecto se basa en las siguientes fases:

- **Búsqueda bibliográfica:** Esta fase básicamente consiste en la búsqueda de artículos científicos, revistas, libros, es decir, toda aquella fuente bibliográfica que pueda proveer información seria, referente al tema de investigación.
- **Clasificación de bibliografía:** Esta segunda fase consiste en clasificar y filtrar la información recolectada del paso anterior, esto con el fin de desechar todo aquel material que no brinde información relacionada con los temas de investigación.
- **Investigación bibliográfica:** Con la bibliografía ya clasificada se procede a obtener información y a relacionarla con los distintos puntos que se encuentran en el esquema de investigación. En esta fase también se realizará procesos que permitan entender el mapeo de los metadatos con las ontologías (clases, propiedades, etc.), de forma manual.
- **Redacción:** Con la información obtenida en el paso anterior se procederá a redactar lo que sería el marco teórico de la investigación.
- **Elección de herramientas:** Con los conocimientos adquiridos se realiza una clasificación y comparación de las distintas herramientas existentes para el desarrollo de la aplicación correspondiente al presente proyecto. Esto implica Repositorios de Objetos de Aprendizaje (ROA), protocolos y estándares de metadatos, ontologías, lenguaje de programación, entorno de desarrollo, etc., de esta manera se facilitará la elección de la más indicada.
- **Diseño:** Una vez seleccionada las herramientas más apropiadas se realiza el diseño de la aplicación, esto implicará los respectivos diagramas que muestren y describan el funcionamiento de la aplicación, los mismos que permitan definir los distintos parámetros para el desarrollo de la aplicación.
- **Implementación:** En esta fase se desarrollará la aplicación para la automatización del proceso de poblado de ontologías, además de la implementación de consultas SPARQL que permitan vincular los resultados con la fuente externa: DBpedia



- **Estudio de caso:** Aquí se describirá un experimento controlado como estudio de caso, el cual servirá para mostrar el funcionamiento de la aplicación desarrollada, en un ambiente controlado.
- **Pruebas:** Esta fase consistirá en realizar pruebas de la aplicación desarrollada, esto implicará realizar la población de distintas ontologías, consultas sobre dichas ontologías y además la generación de vínculos con fuentes externas como DBpedia.
- **Resultados:** Esta última fase consistirá en analizar los resultados obtenidos del experimento controlado y de las pruebas realizadas, con lo cual se buscará evidenciar las ventajas que ofrece la solución implementada como producto final del presente proyecto.

## 1.8 Estructura del proyecto

En esta sección se presenta la estructura del presente proyecto, el mismo que consta de cinco capítulos que detallan el proceso de desarrollo de la solución propuesta. Comenzando con un capítulo introductorio que detalla un panorama general, la problemática del proyecto, justificación, objetivos y la metodología que se manejara en el presente proyecto.

El capítulo 2, “Marco Teórico” presenta una base teórica que permita entender los conceptos generales y necesarios para el desarrollo del presente proyecto, como lo son los objetos de aprendizaje (OA), repositorios de objetos (ROA) y metadatos. Así como también los conceptos que relacionan a la web semántica y a los OA, tales como anotaciones semánticas, ontologías específicas para OA, protocolos de extracción de metadatos y poblado de ontologías. Se ha tomado en consideración los trabajos realizados en los últimos siete años, de tal manera que se pueda establecer un panorama general de los OA dentro del marco de la Web Semántica.

El capítulo 3, “Diseño de la solución” se presentan las explicaciones del proceso de instalación de un repositorio local, necesario para las pruebas de la solución desarrollada que a partir de ahora se la llamara como “Metadata Harvester”, también se presentan en este capítulo explicaciones del proceso de extracción de metadatos mediante el protocolo OAI-PMH, así como también el mapeo de ontologías y los procesos de poblado manual, semiautomático y automático.

El capítulo 4, “Estudio de caso” presenta el desarrollo de un experimento controlado sobre la plataforma de Dspace, bajo la cual se ejecutará la aplicación “Metadata Harvester”. Presentando un escenario que consta de un ROA local, con 46 OA, subidos de forma manual al mismo. Es importante señalar que se trabaja con esta

cantidad de OA debido a que es necesario probar la aplicación en un escenario en el cual estén presentes varios OA al mismo tiempo, y además una búsqueda de OA en el repositorio de Ágora, bajo el criterio de “programación” ha devuelto este número de OA, como se presentara en el desarrollo de este capítulo. El repositorio local permite extraer metadatos de los OA que se encuentre en el mismo y se procede a consumirlos mediante servicios web y bajo el protocolo OAI-PMH, para seguidamente poblar las ontologías, obteniendo como resultado un archivo RDF poblado. Este resultado permite realizar consultas SPARQL sobre este archivo, usando un triple store como Marmotta, y además relacionar estos resultados con fuentes externas como DBpedia. Además, en este capítulo se presentan los resultados del experimento realizado sobre la población de las ontologías, así como también los recursos externos (DBpedia), relacionados con los OA. Y finalmente se realiza un análisis de los resultados, lo cual permitirá evidenciar los beneficios que ofrece la solución desarrollada.

El capítulo 5, “Conclusiones y trabajos futuros” detalla las conclusiones que se han obtenido en el desarrollo del presente proyecto y las líneas de trabajo futuro que se han generado con el mismo.



## Capítulo 2. Marco Teórico

### 2.1 Introducción

El presente capítulo presenta una recopilación de conceptos y temas que pretenden dar un sustento teórico al presente proyecto, razón por la cual se presentan temas como objetos de aprendizaje (OA), repositorios de objetos (ROA) y metadatos. También conceptos que relacionan a la web semántica y a los OA, ontologías, estándares, protocolos y los procesos de poblados de una ontología. Todo esto con el fin de que se pueda establecer un panorama general para el desarrollo correcto del presente proyecto.

### 2.2 Objetos de Aprendizaje

Según la revista de la Universidad Politécnica de Valencia, titulada “Los objetos de aprendizaje como recurso para la docencia universitaria: criterios para su elaboración”, el término Objeto de Aprendizaje, fue introducido por Wayne Hodgins en 1992, mientras trabajaba en el desarrollo de algunas estrategias de aprendizaje. Wayne observó a su hijo jugar con piezas de LEGO y pensó que esta sería una manera fácil y simple para explicar la formación de materiales educativos en pequeñas unidades reutilizables, que permitan el aprendizaje de una forma sencilla. Esta metáfora, conduce a una explicación simple de la reusabilidad de los OA dentro del ámbito de la educación.

Y si bien las definiciones de OA son muy diversas, básicamente se puede decir que un OA es un recurso digital que se crea como apoyo para algún proceso de aprendizaje, y que se orienta a un objetivo específico, y que puede reutilizarse y también puede combinarse con otros OA, lo cual permitirá que pueda cubrir módulos o inclusive cursos completos (Area Moreira & Adell Segura, 2009). Sin embargo, a lo largo del tiempo han surgido varias definiciones de lo que es un OA, donde los autores de las mismas han definido estos conceptos de manera similar. Según Wayne Hodgins un OA es la formación de materiales educativos en pequeñas unidades, que se conectan entre sí y que permiten el aprendizaje de una forma sencilla. Según el LTSC (Learning Technology Standards Committee) o Comité de Estándares de Tecnologías de aprendizaje, los OA se definen como cualquier entidad, digital o no digital, que puede ser utilizada, reutilizada o referenciada durante el aprendizaje apoyado en la tecnología. Según la IEEE (Institute of Electrical and Electronics Engineers), un OA es cualquier entidad, digital o no, que pueda ser utilizada para el aprendizaje, la educación o la capacitación. (Draft Standard for Learning Object Metadata, 2002). David Wiley define a un OA como

cualquier recurso digital que pueda ser reutilizado para apoyar el aprendizaje. Chiappe define a los OA como un conjunto de recursos digitales que puede ser utilizado en diversos contextos, con un propósito educativo y constituido por al menos tres componentes internos: contenidos, actividades de aprendizaje y elementos de contextualización. El centro británico inter-universitario de OA, los define como pedazos de e-learning interactivos basados en la web, y diseñados para explicar un determinado tema de aprendizaje autónomo. Polsani describe al OA como: "Una unidad didáctica, auto contenida e independiente, predispuesta para su reutilización en múltiples contextos instruccionales". Chiappe, Segovia, & Rincon definen a los OA como una "entidad digital, auto contenible y reutilizable, con un claro propósito educativo, constituido por al menos tres componentes internos editables: contenidos, actividades de aprendizaje y elementos de contextualización".

Una definición que engloba las principales características de los OA descritas anteriormente es la planteada por Jorge J. Maldonado, Josefina Siguencia, Juan Pablo Carvallo en el 2015, en su trabajo definen a un OA como: "Una unidad didáctica digital independiente, cuya estructura está formada por un objetivo de aprendizaje específico, un contenido, actividades y una autoevaluación, y que puede ser reutilizada en diferentes contextos tecnológicos y educativos. Además, cuenta con unos metadatos que propicien su localización dentro de los repositorios y permitan abordar su contextualización". (Maldonado J, Siguencia J, & Carvallo J, 2015)

## 2.3 Repositorios de Objetos de Aprendizaje (ROA)

Un ROA, puede ser definido como una plataforma que permite almacenar, buscar, consultar y descargar OA. En otras palabras, son grandes bases de datos o colecciones diseñadas para recopilar OA, y que además cuentan con sistemas de búsquedas de recursos, edición de la información y control de acceso.

Según la revista Electrónica de la Universidad Nacional Autónoma de México un ROA es un dispositivo físico o lógico que permite hacer persistir datos o información. Generalmente es un espacio en un medio de almacenamiento como un disco duro en un servidor, en el que se establece una estructura de carpetas para almacenar los objetos de acuerdo con sus características. Por ejemplo, pueden tenerse carpetas por temas, por tipos de objetos, por nivel educativo, etc. (Peñaloza & Durán, 2008)

### 2.3.1 Ágora (Asistencia para la Gestión de Objetos Reusables de Aprendizaje.)

Agora es un ROA y una propuesta para mejorar la gestión de los OA mediante un enfoque de asistencia en la ejecución y control de las actividades en los OA. Es un entorno basado en servicios, que integra las características de almacenamiento, edición y publicación de OA y presenta un área de trabajo donde cada usuario puede acceder a todas las herramientas que ofrece la plataforma y modificar su perfil o administrar sus OA. Además, en AGORA, el usuario interactúa con una aplicación Web para invocar los servicios expuestos por la plataforma. (Menéndez D., Castellanos B., & Zapata G., 2012) Todo esto permite la interacción con los OA almacenados en este ROA, como por ejemplo la imagen 2.1 presenta algunos de los resultados que devuelve una búsqueda de OA referentes al tema de “programación”, esto mediante el buscador incorporado en el repositorio AGORA.

Algunas características importantes de este repositorio son los siguientes (Menéndez D. et al., 2012):

- AGORA se basa en un modelo de gestión que hace hincapié en registrar toda la información generada como resultado de la ejecución de las actividades de desarrollo de un OA, con el propósito de analizar y extraer normas y reglas en el uso de la aplicación por los usuarios.
- AGORA es un entorno basado en servicios, que integra características de almacenamiento, edición, publicación de OA.
- AGORA permite la interacción entre el usuario y una aplicación Web para invocar los servicios que están en la plataforma.
- AGORA utiliza como interfaz de usuario, el concepto de "escritorio". Esto para representar el área de trabajo donde se puede acceder a todas las herramientas que ofrece la plataforma y modificar su perfil o administrar sus OA.
- La plataforma utiliza los formatos de archivo más comunes para OA (ppt, pdf, doc, jpg, zip). Incluye también un generador de metadatos de OA que simplifica la creación de los mismos, así como varias herramientas de comunicación para colaborar o compartir información con otros usuarios de la plataforma”.
- AGORA ofrece una información completa de los metadatos de sus OA.



Imagen. 2.1 Repositorio Ágora. Búsqueda de OA referentes a programación

## 2.3.2 MERLOT (Multimedia Educational Resource for Learning and Online Teaching)

Es un repositorio centralizado que contiene sólo metadatos y apunta a los objetos ubicados en sitios remotos. Este ROA es independiente y funciona como un portal de OA. (López Guzmán & García Peñalvo, 2009). Además, este ROA también permite la evaluación de OA, donde el criterio utilizado es la evaluación individual y la remisión a las normas de los OA. Está basado en tres dimensiones: calidad del contenido, efectividad potencial y facilidad de uso, cada una evaluada mediante una escala de cinco valores donde una estrella denota que el material no es digno de uso y la máxima puntuación indica que el OA posee excelencia en todo. (Haughey & Muirhead, 2005) (Rincón Valadez, Martínez Lazcano, & Curiel Anaya, 2012). La imagen 2.2 presenta la interfaz de búsqueda en MERLOT, donde se presente algunos de los resultados que devuelve este ROA, al realizar una búsqueda de OA bajo el criterio de “programación”.



Search > MERLOT Results

Current Search:  
Keywords: programacion

Refine Your Search:

- Materials (24)
- Members (2)
- Learning Exercises (1)
- Comments (27)
- Bookmark Collections (8)
- Course ePortfolios (2)
- Peer Reviews (1)

Materials 1 - 10 shown of 24 results

**Programacion de S7 1200**

El video muestra la programacion del nuevo PLC S7 1200 y su comparacion con la version anterior S7 200  
Material Type: Presentation  
Author: Siemens Siemens  
Date Added: Nov 11, 2011 Date Modified: Nov 11, 2011

**Programación con Swing en JAVA**

Archivo PDF que contiene información sobre los controles Swing de JAVA, tanto nombre como propiedades y comportamientos de... see more  
Material Type: Tutorial  
Author: Francisco Calceño  
Date Added: Nov 17, 2012 Date Modified: Oct 27, 2015

**Tipos de Lenguajes de Programación**

describe los diversos lenguajes de programación existentes en el mercado.  
Material Type: Tutorial  
Author: Unknown  
Date Added: Jan 13, 2012 Date Modified: Sep 12, 2018

See More Materials

Bookmark this material

About this material:

Editor Review (not reviewed)  
User Rating (not rated)  
Discussion (none)  
In (1) Bookmark Collections  
Course ePortfolios (none)  
Learning Exercises (none)  
Accessibility Info (none)

Bookmark this material

About this material:

Editor Review avg: ★★★★★  
User Rating (not rated)  
Discussion (none)  
In (2) Bookmark Collections  
Course ePortfolios (none)  
Learning Exercises (none)  
Accessibility Info (none)

Bookmark this material

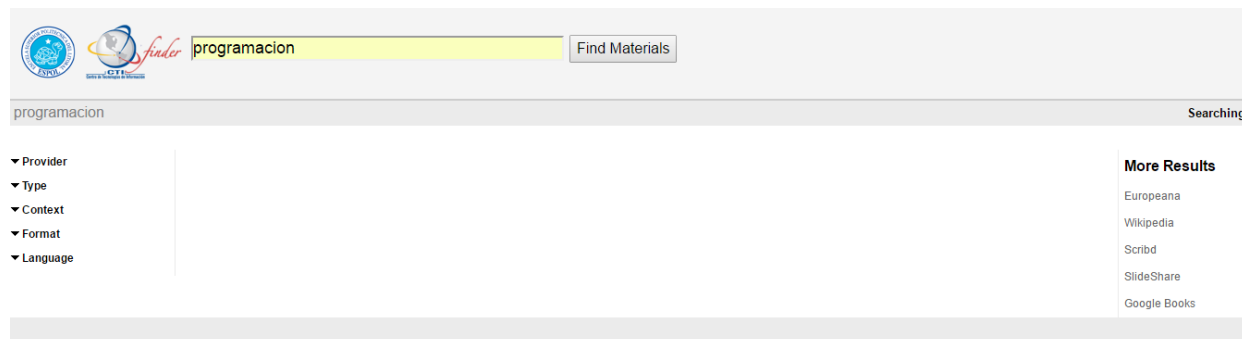
About this material:

Editor Review (not reviewed)  
User Rating ★★★★★  
Discussion (2 Comments)  
In (2) Bookmark Collections  
Course ePortfolios (none)  
Learning Exercises (none)  
Accessibility Info (none)

Imagen. 2.2 Repositorio Merlot. Búsqueda de recursos referentes a programación.

### 2.3.3 Plataforma ARIADNE

Es un software desarrollado por la fundación ARIADNE (Alliance of Remote Instructional Authoring & Distribution Networks for Europe) que permite la edición, clasificación, almacenamiento y consulta de recursos en línea. Como repositorio ARIADNE es el encargado de almacenar y manejar OA y sus metadatos relacionados. Su desarrollo está basado en estándares y especificaciones como metadatos LOM, recolección OAI-PMH, y consultas SQL. En la imagen 2.3 se presenta la interfaz principal del software ARIADNE, denominada Ariadne Finder.



Logo of Ariadne Finder

programacion Find Materials

programacion Searching.

Provider  
Type  
Context  
Format  
Language

More Results

- Europeana
- Wikipedia
- Scribd
- SlideShare
- Google Books

Imagen. 2.3 Interfaz de Ariadne Finder

Una gran limitante de las herramientas ARIADNE es la falta de documentación y la necesidad de contar con las versiones exactas del software requerido, pues están diseñadas para ciertas versiones y con otras simplemente no funcionan (Cajal, Moraga, & Rosanigo, 2012)

### 2.3.4 Tabla comparativa de ROA

A continuación, se presenta una tabla comparativa de los ROA mencionados anteriormente, donde se puede ver que Ágora ofrece ciertos beneficios con respecto a los demás. Además de la amplia documentación que esta ofrece.

	Ágora	Merlot	Ariadne
<b>Nivel</b>	Superior	Superior	Superior
<b>Organización/País</b>	Internacional  México, Centroamérica y el Caribe  Profesores de la Universidad de Castilla-La Mancha, en España.	Internacional	Internacional
<b>Acceso</b>	Con registro de usuario	Abierto  Registro de usuario para poder contribuir	Limitado
<b>Metadatos</b>	Si	No	No
<b>OA</b>	Distribuidos	Distribuidos	Limitados
<b>Gestión de Usuarios</b>	Si	Si	No
<b>OA empaquetados</b>	ppt, pdf, doc, jpg, zip	No	No

Documentación	Si	Escaza	Limitada
---------------	----	--------	----------

Tabla 2.1 Comparación de los ROA. Fuente: autores

## 2.4 Metadatos

Con el paso de los años el concepto de metadato se ha popularizado mucho ya que la información en la Web ha llegado a crecer tanto y está necesita ser organizada. Esta información podría aprovecharse de una manera óptima, permitiendo que los sistemas inter operen, sin importar su campo de aplicación. Justamente hacia esto apunta la Web Semántica, la cual consiste en adicionar metadatos a los recursos, lo que permite describir el contenido, significado y relación con otros recursos. Por lo tanto, los metadatos consisten en un conjunto de atributos o elementos necesarios para describir un OA, lo cual permite la catalogación digital de la información del mismo, esto para su reutilización en diversos contextos. Además, los metadatos permiten potenciar a los OA como recursos educativos recuperables, localizables, intercambiables y reutilizables. A través de ellos se puede llevar a cabo el primer acercamiento con el objeto y conocer rápidamente sus principales características. (Berlanga, López, Morales, & García, 2011)

### 2.4.1 Estándares de metadatos

Actualmente existen varias iniciativas de estandarización de metadatos, siendo LOM y Dublin Core los más importantes. Estos estándares permiten describir un OA siguiendo determinadas normas y reglas establecidas.

#### 2.4.1.1 LOM

Dentro del ámbito educativo y para metadatos de propósito específico, destaca el estándar LOM, el cual está orientado a OA y plataformas educativas. Por esta razón este es un estándar por excelencia para Metadatos de OA. (Agudelo Benjumea, 2008). Un ejemplo de aplicación de este estándar es presentado en el trabajo titulado: “LOM-ES: Un perfil de aplicación de LOM”, donde los autores describen un perfil de aplicación de LOM, con la finalidad de adecuarlo a la realidad y problemática del sistema educativo español no universitario. Además, se presenta la aplicación de LOM-ES, en un ámbito de un programa de Internet en el aula, mediante la creación de OA y ROA. (Canabal, Sarasa, & Sacristán, 2006)

Este estándar está patrocinado por el Comité de Estandarización de Tecnologías Educativas del IEEE, define y especifica un esquema de metadatos que permite múltiples implementaciones, atributos, definiciones, una estructura jerárquica que los relaciona entre ellos, aspectos teóricos del esquema, etc., pero no incluye



información acerca de cómo representar estos metadatos o con qué mecanismos se puede transmitir y procesar esta meta información.

Los metadatos LOM también cubren aspectos pedagógicos como el estilo de enseñanza o el nivel de interactividad. El estándar de IEEE 1484.12.1 del año 2002 especifica los metadatos para objetos de aprendizaje. La norma define el modelo de datos y los vocabularios usados como dominios. Como puede verse expresado en el siguiente mapa, LOM se divide en 9 categorías (Temesio, 2015)

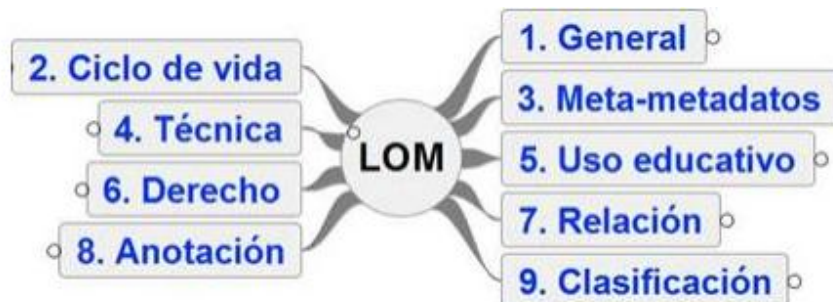


Imagen. 2.4 Categorías LOM. Fuente: <https://goo.gl/fRdaf5>

Donde cada categoría indica el área de descripción de un recurso educativo (Marzal García-Quismondo, Calzada Prado, & Cuevas Cerveró, 2006):

- **General.** Recoge algunos de los principales elementos de identificación del documento descrito: código de identificación, título del recurso, idioma, breve descripción de su contenido, palabras clave, cobertura temporal o geográfica, estructura e información sobre su granularidad o nivel de agregación.
- **Ciclo de vida.** Recoge información relativa a la autoría, fecha de creación, versión y estado del recurso descrito.
- **Meta-metadatos.** Proporciona información sobre el esquema de metadatos empleado en la descripción del recurso, fecha, nombre del creador e idioma del registro.
- **Técnica.** Recoge información relativa al formato, tamaño, URI, duración y requisitos técnicos para la utilización del recurso.
- **Uso educativo.** Describe el uso educativo del recurso: tipo de recurso de que se trata, tipo y nivel del usuario al que se dirige, contexto de utilización, tipo y nivel de interactividad que presenta, densidad semántica, dificultad, idioma y descripción de su uso.



- **Derechos.** Recoge aspectos relativos a las restricciones de uso asociadas al recurso: coste, protección de los derechos de autor y otras restricciones de uso.
- **Relación.** Proporciona información sobre las relaciones, en caso de que las haya, establecidas entre el recurso descrito y otros recursos.
- **Anotación.** Recoge los comentarios del catalogador sobre el uso pedagógico del recurso.
- **Clasificación.** Descripción del contenido del recurso a partir de uno o varios sistemas de clasificación, vocabularios y palabras clave.

Para cada una de esas categorías existen varios elementos que corresponden a un aspecto general que constituye la categoría.

En total, LOM tiene 77 elementos contenidos en las 9 categorías presentadas en el mapa de la imagen 2.4. Debido a lo extenso que es describir todos estos elementos, en el presente trabajo se presenta a modo de ejemplo el detalle de una de las 9 categorías, en este caso la categoría: Uso Educativo, el cual se ve en la tabla 2.2:

Elemento	Descripción y uso
5.1 Tipo de Interactividad	Tipo de aprendizaje predominante soportado por este objeto de aprendizaje: activo, expositivo y combinado
5.2 Tipo de recurso humano	Tipo de recurso
5.3 Nivel de Interactividad	Grado de interactividad que caracteriza al objeto educativo (referida al grado en que el aprendiz puede influir en el aspecto o comportamiento del objeto educativo).
5.4 Densidad Semántica	Grado concisión de un objeto educativo. Puede ser estimada en función de su tamaño, ámbito y duración, e independientemente de su dificultad.
5.5 Destinatario	Usuario principal para el que ha sido diseñado el objeto educativo: profesor, autor, aprendiz, administrador.

5.6 Contexto	Entorno principal en el que se utilizará el objeto educativo: Escuela, Educación, Secundaria, Entretenimientos y otros.
5.7 Rango Típico de Edad	Edad del destinatario, por rangos de edades.
5.8 Dificultad	Grado de dificultad que presenta, para los destinatarios típicos, trabajar con y utilizar este objeto educativo.
5.9 Tiempo Típico de Aprendizaje	Tiempo medio de utilización del recurso.
5.10 Descripción	Comentarios sobre cómo debe utilizarse este objeto educativo.
5.11 Idioma	Idioma utilizado en el objeto educativo

Tabla 2.2 Elementos de tipo educativo en LOM

Además, es importante señalar que los elementos de una categoría pueden ser obligatorios, opcionales o recomendados, tal es el caso que se presenta a continuación para la categoría General:



Imagen. 2.5 Elementos de categoría General.

Fuente: <https://goo.gl/fRdaf5>

- Identificador (obligatorio)
- Título (obligatorio)
- Idioma (obligatorio)

- Descripción (obligatorio)
- Palabra clave (recomendado)
- Ámbito (opcional)
- Estructura (opcional)
- Nivel de agregación (obligatorio)

### 2.4.1.2 Dublin Core

A diferencia de LOM, Dublin Core es un estándar de criterio más general, y es muy utilizado en muchos ámbitos, entre ellos el bibliotecario y documental. El estándar Dublin Core establece metadatos para describir, identificar, procesar, encontrar y recuperar un documento introducido en la red. Esto facilita la indexación automática y mejora la efectividad de los motores de búsqueda sobre cualquier recurso descrito con Dublin Core.

El conjunto de elementos de metadatos Dublin Core está previsto para describir documentos. Estos metadatos poseen etiquetas descriptivas que pretenden transmitir un significado semántico y promover una interoperabilidad con todos los recursos de la red. Algo importante que se debe mencionar, es que cada elemento es opcional, puede repetirse y pueden aparecer en cualquier orden. Además, se puede clasificar estos elementos en tres grupos que indican la clase o el ámbito de la información que se guarda en ellos: (Red Iris, 2011), esta clasificación se presenta en la tabla 2.3.

Clasificación en base a la clase y al ámbito		
Contenido	Propiedad Intelectual	Instanciación
Title	Creator	Date
Subject	Publisher	Type
Description	Contributor	Format
Source	Rights	Identifier
Lenguaje		
Relation		
Coverage		

Tabla 2.3 Grupos principales del estándar Dublin Core. Fuente: autores

Elementos relacionados con el contenido del recurso:

- **Title** (título)
- **Subject** (tema)
- **Description** (descripción)
- **Source** (fuente)
- **Lenguaje** (lenguaje)
- **Relation** (relación)
- **Coverage** (cobertura).

Elementos relacionados con el recurso cuando es visto como una propiedad intelectual:

- **Creator** (autor)
- **Publisher** (editor) y, otras colaboraciones
- **Contributor** (otros autores/colaboradores)
- **Rights** (derechos).

Elementos relacionados con la instanciación del recurso:

- **Date** (fecha)
- **Type** (tipo de recurso)
- **Format** (formato)
- **Identifier** (Identificador)

A continuación se presenta una pequeña descripción de los metadatos de este estándar (Metadata Innovation Dublin Core, 2012):

Etiqueta del elemento DC.	Descripción
DC. Title	Título: El nombre dado a un recurso. Típicamente, un título es el nombre formal por el que es conocido el recurso.
DC. Creator	Autor: La entidad responsable de la creación del contenido intelectual del recurso. Por ejemplo: una persona, una organización o un servicio. Típicamente, el nombre del creador podría usarse para indicar la entidad.

DC. Subject	Materias y palabras clave: El tema del contenido del recurso. Un tema será expresado como palabras clave, frases clave o códigos de clasificación que describan el tema de un recurso.
DC. Description	Descripción: La descripción del contenido del recurso puede incluir, pero no se limita a un resumen, tabla de contenidos, referencia a una representación gráfica de contenido o una descripción de texto libre del contenido.
DC. Publisher	Editor: La entidad responsable de hacer que el recurso se encuentre disponible. Ejemplos de editores son una persona, una organización o un servicio.
DC. Contributor	Colaborador. La entidad responsable de hacer colaboraciones al contenido del recurso. Ejemplos de colaboradores son una persona, una organización o un servicio.
DC. Date	Fecha: Una fecha asociada con un evento en el ciclo de vida del recurso. Típicamente, la fecha será asociada con la creación o disponibilidad del recurso.
DC. Type	Tipo: La naturaleza o categoría del contenido del recurso. El tipo incluye términos que describen las categorías generales, funciones, géneros o niveles de agregación del contenido.
DC. Format	Formato: La manifestación física o digital del recurso. El formato puede incluir el tipo de media o dimensiones del recurso.
DC. Identifier	Identificación: Una referencia no debe ser ambigua para el recurso y debe ser la que lo identifique dentro de un contexto dado.
DC. Source	Fuente: Una referencia a un recurso del cual se deriva el recurso actual. El recurso actual puede derivarse, en todo o en parte, de un recurso fuente.
DC. Language	Lengua: La lengua del contenido intelectual del recurso. Se define las etiquetas de dos y tres letras primarias para lenguaje, con subetiquetas opcionales. Ejemplo: "en" u "eng" para inglés, "akk" para Acadio, y "en-GB" para inglés usado en Reino Unido.

DC. Relation	Relación: Una referencia a un recurso relacionado. Se recomienda referenciar el recurso por medio de una cadena de números de acuerdo con un sistema de identificación formal.
DC. Coverage	Cobertura: La extensión o ámbito del contenido del recurso. La cobertura incluiría la localización espacial (un nombre de lugar o coordenadas geográficas), el período temporal (una etiqueta del período, fecha o rango de datos) o jurisdicción (tal como el nombre de una entidad administrativa).
DC. Rights	Derechos: La información sobre los derechos de propiedad y sobre el recurso. Este elemento podrá contener un estamento de gestión de derechos para el recurso, o referencia a un servicio que provea tal información. La información sobre derechos a menudo corresponde a los derechos de propiedad intelectual, copyright y otros derechos de propiedad.

Tabla 2.4 Elementos de Dublin Core. Fuente: <https://goo.gl/pr7nKM>

### 2.4.1.3 SCORM (Sharable Content Object Reference Mode)

Es un conjunto de estándares y especificaciones que permite crear objetos pedagógicos estructurados. Este estándar define un modelo de agregación de contenidos para el aprendizaje a través de Internet. Este es un modelo de referencia que proporciona un conjunto de especificaciones y guías que permiten elaborar cursos que cumplan los requerimientos de la formación a través de Internet. (Miguel Rebollo Pedruelo, 2004)

#### Características de SCORM

- SCORM normaliza y especifica, las necesidades más interesantes para la interoperabilidad de los sistemas de formación virtual.
- Define todos los datos (alumnos, organización, actividad formativa, resultados de aprendizaje, datos de evaluación, etc.) que debe recoger una plataforma educativa.
- Define y especifica todos los datos que puede generar el alumno en su navegación y aprendizaje por un curso SCORM compatible (teóricos, prácticos, evaluaciones, etc.).
- Define cómo debe construirse un curso SCORM compatible, su estructura, sus posibles itinerarios pedagógicos, restricciones y exigencias, etc., y como calificar con metadatos los objetos educativos.

- Define cómo se debe importar/exportar un curso SCORM compatible entre cualquier utilidad de creación de contenidos.

#### 2.4.1.4 IMS (Instruction Management Systems)

IMS es un consorcio que agrupa a vendedores, productores, implementadores y consumidores de e-learning. Este se enfoca completamente a desarrollar especificaciones en formato XML. (Hernández, 2001). Sus especificaciones cubren un amplio rango de características, que van desde los metadatos, la interoperabilidad de intercambiar el diseño instruccional entre plataformas, hasta la creación de cursos online para alumnos que tengan alguna discapacidad visual, auditiva u otra. Todas las especificaciones de IMS se detallan en tres documentos que son: (Hilera González & Hoya Marín, 2010)

**Guía de Implementación y Consejos.** Es el documento más narrativo de los tres. En este documento están incluidos: la forma de uso de la especificación, la relación con otras especificaciones, y cualquier tipo de información variada que pueda servir de ayuda.

**Modelo de Información.** Documento que muestra la estructura de datos completa. Posee una tabla detallada de cada uno de los elementos de la especificación.

**Documento de enlace.** Documento que ofrece la forma de representar la estructura de datos de la especificación en XML.

### 2.5 Extracción de metadatos mediante servicios web

Los servicios web son un conjunto de tecnologías que tienen la capacidad de inter-operar en la Web.

Básicamente lo que hacen es:

- Proporcionar mecanismos de comunicación estándar entre diferentes aplicaciones.
- Permiten exponer y hacer disponibles funcionalidades (servicios) de los sistemas informáticos de las organizaciones mediante tecnologías y protocolos Web estándar.
- Actúan como componentes independientes que se pueden integrar para formar sistemas distribuidos complejos.

Además, es posible acceder a distintos ROA mediante servicios web, con el propósito de extraer los metadatos de los OA. Se realiza este proceso mediante

protocolos de extracción de metadatos, entre ellos está uno específico, llamado OAI-PMH.

## 2.5.1 Protocolos para la extracción de metadatos de los repositorios.

Los protocolos son un conjunto de normas que regulan el intercambio de información entre ordenadores. Son reglas de comunicación de común acuerdo entre computadoras que transfieren información conectadas a Internet. (Benítez & Robayo, 2006)

### 2.5.1.1 Protocolo Z39.50

Este protocolo permite comunicar sistemas que funcionan en distinto hardware y usan distinto software. Fue diseñado para solucionar los problemas asociados a la búsqueda en múltiples bases de datos con diferentes lenguajes y procedimientos.

Básicamente, este es un protocolo para la recuperación de información que se basa en la estructura cliente/servidor y que facilita la interconexión de sistemas informáticos. Un ámbito importante de uso de este protocolo es la educación, ya que el mismo hace posible la comunicación entre sistemas bibliotecarios o documentales que utilizan diferentes plataformas software y hardware. También permite hacer búsquedas simultáneas en distintos catálogos o bases de datos utilizando una única interfaz de usuario o consulta, recuperar la información, ordenarla y exportar los registros bibliográficos obtenidos. (Benítez & Robayo, 2006)

Según Aldo Guajardo Salinas, la función esencial de cualquier cliente es permitir al usuario realizar búsquedas en bases de datos que cuenten con un servidor. Y al realizar una búsqueda con el protocolo Z39.50 existen los siguiente servicios que se agrupan en lo que el estándar denomina "facilidades": (Guajardo Salinas, 2010).

**Inicialización:** Configuración de las reglas de comunicación y negociación.

**Búsqueda:** Envío de una cadena de búsqueda a una base de datos y recuperación de un conjunto de resultados.

**Recuperación:** Recuperación de registros especificados en el cliente.

**Borrar resultados:** Permite a un cliente solicitar que se borre un conjunto de resultados.



**Control de acceso:** Proceso de verificación por parte del servidor, por ejemplo: pedir passwords y verificar.

**Control de cuentas:** Es el proceso de gestión de cuentas, lo que permite manejarlas de una manera adecuada.

**Ordenar:** Permite que el cliente solicite al servidor un ordenamiento del conjunto de resultados.

**Visualizar índices:** Se refiere al proceso de índice de resultados en el servidor.

**Servicios extendidos:** Permite al cliente realizar unas tareas extendidas.

**Terminación:** Concluye la negociación y la conexión del protocolo.

### 2.5.1.2 Protocolo OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting)

El protocolo OAI-PMH se creó con la misión de desarrollar y promover estándares de interoperabilidad que faciliten la difusión eficiente de contenidos en internet, donde los metadatos a transmitir mediante estos deberán codificarse bajo algún estándar, como LOM o Dublin Core. Este protocolo básicamente utiliza servicios basados en los estándares HTTP para emitir preguntas y obtener respuestas entre un servidor y un cliente, donde este puede pedir al primero que le envíe metadatos según determinados criterios como por ejemplo la fecha de creación de los datos y donde como respuesta, el primero devuelve un conjunto de registros en formato XML, incluyendo identificadores de los objetos descritos en cada registro. Las peticiones se emiten utilizando los métodos get o post del protocolo HTTP y constan de una lista de opciones con la forma de pares del tipo clave=valor. Según el artículo “Protocol for metadata harvesting (OAI-PMH)” Existen seis peticiones que un cliente puede realizar a un servidor: (Barrueco & Subirats Coll, 2013), los cuales son los siguientes:

**GetRecord.** Recupera un registro concreto y necesita dos argumentos: identificador del registro pedido y especificación del formato bibliográfico en que se debe devolver.

**Identify.** Utilizado para recuperar información sobre el servidor: nombre, versión del protocolo que utiliza, dirección del administrador, etc.

**ListIdentifiers.** En lugar de los registros completos recupera sólo sus encabezamientos.

**ListRecords.** Igual que el anterior, pero recupera los registros completos.

**ListSets.** Hace posible la recuperación de un conjunto de registros, los cuales son creados opcionalmente por el servidor para facilitar una recuperación selectiva de los registros. Sería una clasificación de los contenidos según diferentes entradas. Un cliente puede pedir que se recuperen sólo los registros pertenecientes a una determinada clase. Los conjuntos pueden ser simples listas o estructuras jerárquicas.

**ListMetadataFormats.** Devuelve la lista de formatos bibliográficos que utiliza el servidor.

Por lo tanto, el protocolo OAI-PMH es una interfaz que permite el acceso a los metadatos provenientes de distintas fuentes, plataformas y repositorios de contenidos, los mismos que devuelven metadatos en formato XML para su utilización mediante este protocolo. Además, soporta múltiples formatos y estándares para expresar los metadatos.

La imagen 2.6 muestra las peticiones descritas anteriormente:

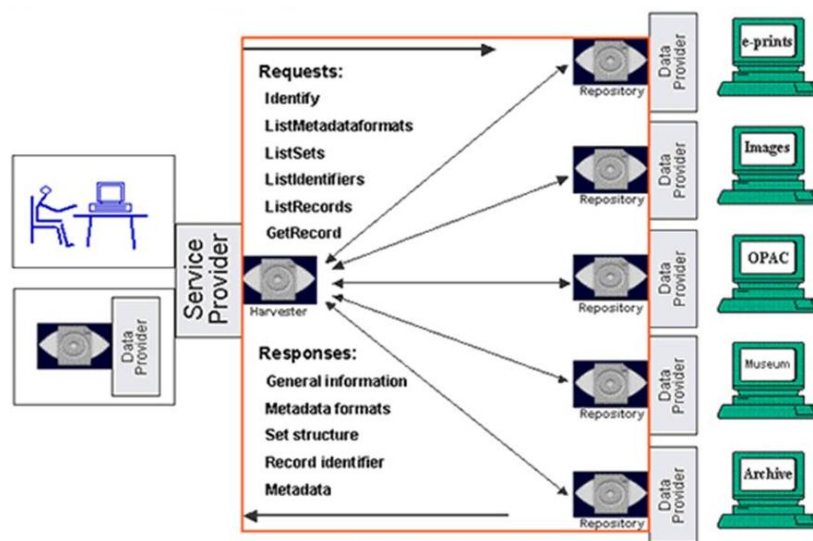


Imagen. 2.6 Peticiones en el protocolo OAI-PMH. Fuente: <https://goo.gl/6RWNup>

La llamada a un servicio web mediante este protocolo es como se muestra en la imagen siguiente:



Imagen. 2.7 Llamada a servicio web usando el protocolo OAI-PMH. Fuente: autores

En la imagen anterior se observa que es necesaria la especificación del protocolo con el cual se está trabajando. Así mismo, es necesario enviar los parámetros para la extracción de los metadatos, mediante palabras reservadas denominadas verb los mismos que dependerán de la información que se va a extraer, en el caso de la imagen 2.8 se está extrayendo información de los grupos de un repositorio. Existen otros casos en los cuales los parámetros de consulta serán los siguientes de acuerdo a la información que se necesite:

- **Extracción de colecciones o grupos de recursos:**

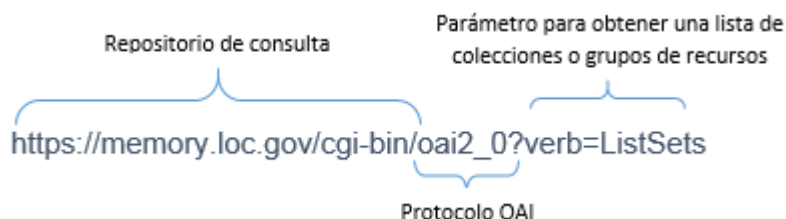


Imagen. 2.8 Extracción de colecciones o grupos de recursos. Fuente: autores

- **Extracción de metadatos de un ROA (Library of Congress Open Archive Initiative Repository 1):**



Imagen. 2.9 Servicio web que permite la extracción de metadatos del Library of Congress Open Archive Initiative Repository. Fuente: autores

- **Extracción de formatos de metadatos:**

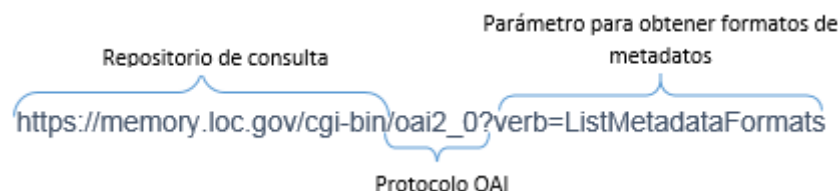


Imagen. 2.10 Servicio web que permite la extracción de formatos de metadatos. Fuente: autores

- **Extracción de un conjunto de metadatos (referentes a la música)**



Imagen. 2.11 Servicio web que permite la extracción de un conjunto de metadatos referentes a la música. Fuente: autores

- **Extracción de un conjunto de identificadores bajo el prefijo: oai\_dc**

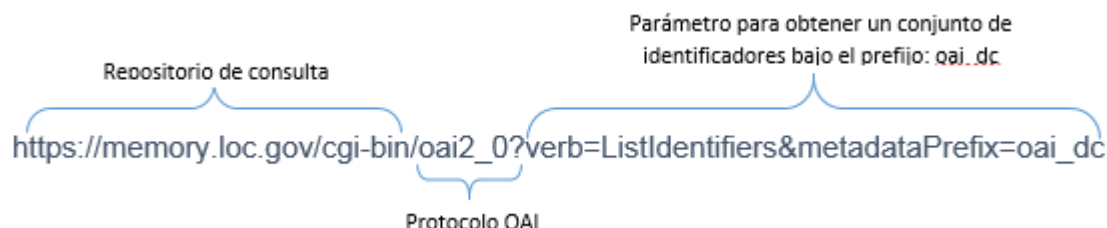


Imagen. 2.12 Servicio web que permite la extracción de un conjunto de identificadores por el prefijo oai\_dc. Fuente: autores

- **Extracción del conjunto de metadatos de un OA, bajo el estándar Dublin Core:**

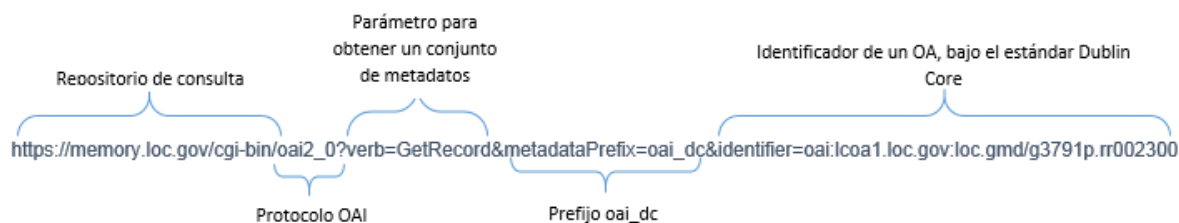


Imagen. 2.13 Servicio web que permite la extracción de los metadatos de un O, por el prefijo oai\_dc y por un identificador. Fuente: autores

En este último caso es necesaria la especificación del identificador del OA para la extracción de sus metadatos. Este servicio Web devuelve como resultado lo siguiente:

```
<?xml version="1.0" encoding="UTF-8" ?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2016-11-07T05:45:39Z</responseDate>
  <request verb="GetRecord" identifier="oai:local.gov:loc.gmd/g3791p.rr002300" metadataPrefix="oai_dc">http://memory.loc.gov/cgi-bin/oai2_0</request>
  <GetRecord>
    <record>
      <header>
        <identifier>oai:local.gov:loc.gmd/g3791p.rr002300</identifier>
        <timestamp>2005-11-21T17:00:59Z</timestamp>
        <setSpec>gmd</setSpec>
      </header>
      <metadata>
        <?xml version="1.0" encoding="UTF-8" ?>
        <dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/ http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
          <dc:title>
            New railroad map of the state of Maryland, Delaware, and the District of Columbia. Compiled and drawn by Frank Arnold Gray.
          </dc:title>
          <dc:creator>Gray, Frank Arnold.</dc:creator>
          <dc:subject>Railroads--Middle Atlantic States--Maps.</dc:subject>
          <dc:description>
            Shows drainage, canals, stations, cities and towns, counties, canals, roads completed, narrow gauge and proposed railroads with names of lines. Includes list of railroads.
          </dc:description>
          <dc:description>Scale 1:633,600.</dc:description>
          <dc:description>LC Railroad maps, 230</dc:description>
          <dc:description>Description derived from published bibliography.</dc:description>
          <dc:publisher>Philadelphia</dc:publisher>
          <dc:date>1876</dc:date>
          <dc:type>Image</dc:type>
          <dc:type>map</dc:type>
          <dc:type>cartographic</dc:type>
          <dc:identifier>http://hdl.loc.gov/loc.gmd/g3791p.rr002300</dc:identifier>
          <dc:language>eng</dc:language>
          <dc:coverage>United States--Middle Atlantic States</dc:coverage>
        </dc:dc>
      </metadata>
    </record>
  </GetRecord>
</OAI-PMH>
```

Imagen. 2.14 Resultado de la consulta de la imagen 2.20. Extracción de metadatos de un OA. Fuente: autores

Es importante notar que el protocolo OAI-PMH está diseñado para trabajar sobre varios estándares, siendo Dublin Core el más utilizado. Razón por la cual los metadatos que se extraigan mediante este protocolo, estarán sobre dicho estándar.

Una aplicación práctica de las peticiones mostradas en las imágenes 2.7, 2.8, 2.9, 2.10, 2.11, 2.12, 2.13 se presentará en el estudio de caso del capítulo 4.

## 2.6 Objetos de Aprendizaje dentro del marco de la Web Semántica

El surgimiento de la Web Semántica permite muchos cambios en la educación on-line, por ejemplo, la función de adaptabilidad automática a las necesidades específicas del usuario. Dentro del campo de la educación, esto permite coleccionar datos del alumno y crear un modelo del estudiante, identificar estudiantes con resultados destacados, etc. (del Moral & Cernea, 2005). En este ámbito el concepto de OA juega un papel importante aplicaciones modernas de e-learning ya que se han transformado en uno de los elementos principales de varios LMS (Learning Management System) y LCM (Learning Content Management System)(Han & Gao, 2012)

## 2.6.1 Anotaciones Semánticas

Se puede considerar a una anotación como la información sobre entidades o conceptos de una ontología, o también las referencias que hay en un texto sobre un repositorio semántico, las cuales establecen una relación entre los contenidos, como una imagen o palabra y una URI (Uniform Resource Identifier) tomada de una ontología o de otra fuente de conocimiento como se presenta en la figura 2.15.

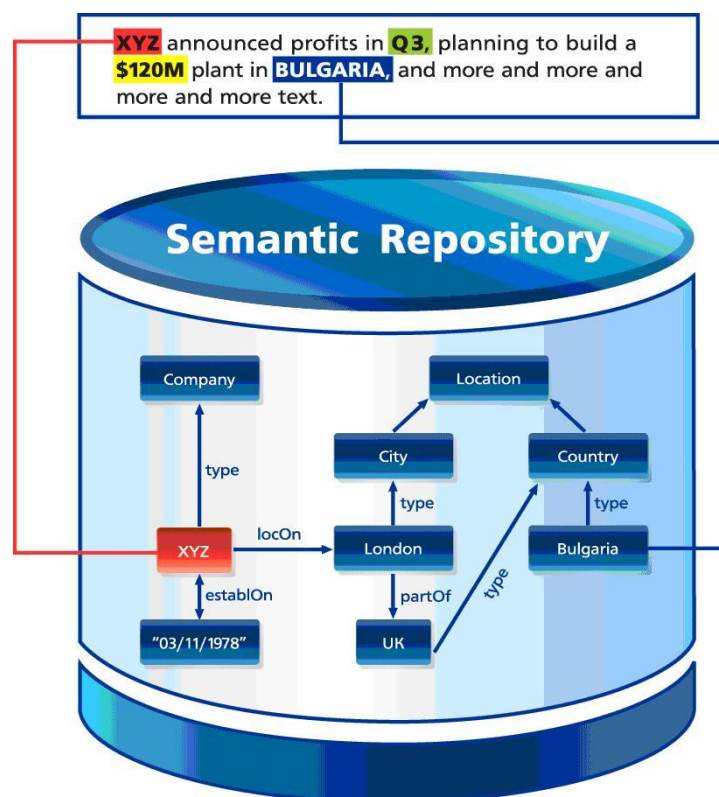


Imagen. 2.15 Anotación Semántica a una Ontología. Fuente Semantic Annotation, Indexing, and Retrieval

### 2.6.1.1 Lenguaje OWL (Ontology Web Language)

Es un lenguaje de marcado codificado en XML que sirve para la publicación de ontologías en la www (World Wide Web) a partir de un vocabulario más amplio y una sintaxis más fuerte que la que permite RDF (Resource Description Framework o Marco de Descripción de Recursos). Además, el lenguaje OWL apoya el intercambio y la reutilización de ontologías, por lo que se permite a una ontología importar de otra ontología. Cuando una ontología importa de otra, todas las clases,

propiedades y definiciones individuales que tiene la ontología importada, están disponibles para su uso en la ontología que importa.

## 2.6.2 Ontologías basadas en estándares de metadatos

El incremento en la cantidad de OA que existen actualmente en internet ha traído consigo problemas para su correcto manejo y acceso. Los OA cuentan con un conjunto de metadatos, los cuales brindan información sobre su contenido y contexto. Estos metadatos han sido usados para mejorar su manejo dentro de distintos ROA. Sin embargo, en la actualidad se han presentado inconvenientes con el manejo de la gran cantidad de OA existentes, además de la descripción limitada que ofrece los metadatos, la cual no puede adaptarse por sí solas a las nuevas tecnologías web (Ouafia, Hélène Abel, & Moulin, 2008). Es así que durante los últimos años se han realizado distintos trabajos con la finalidad de mejorar la descripción de los OA mediante el uso de ontologías.

### 2.6.2.1 LOM2OWL (estándar LOM)

Dentro del marco de la Web semántica, se define una ontología para el modelo de descripción de objetos de aprendizaje IEEE LOM. Esta ontología, denominada LOM2OWL, se implementa en el lenguaje OWL.

En el trabajo titulado “Una ontología en OWL para la representación semántica de objetos de aprendizaje”, sus autores describen el proceso de desarrollo de la ontología LOM2OWL (Fermoso García, Sánchez Alonso, & A. Sicilia, 2008). Debido a que el estándar IEEE LOM utiliza una definición jerárquica para describir los metadatos de un OA, la definición de la ontología LOM2OWL cuenta con una estructura similar a la estructura jerárquica del estándar. La representación ontológica parte de la clase básica *learningObject* en el nivel jerárquico más alto, la cual se relaciona con otras clases que representan categorías y subcategorías del estándar LOM. Adicionalmente LOM2OWL implementa distintas *object property* y *data property* para especificar las relaciones entre clases u objetos y sus propiedades. En la imagen 2.16 se puede observar todas las relaciones directas de la clase *learningObject*. Por ejemplo, la clase *learningObject* se vincula con la clase *lomThecnicalRequirement*, la cual hace referencia a la categoría *Técnica* del estándar LOM, mediante la propiedad *hasTechnicalRequirement*.



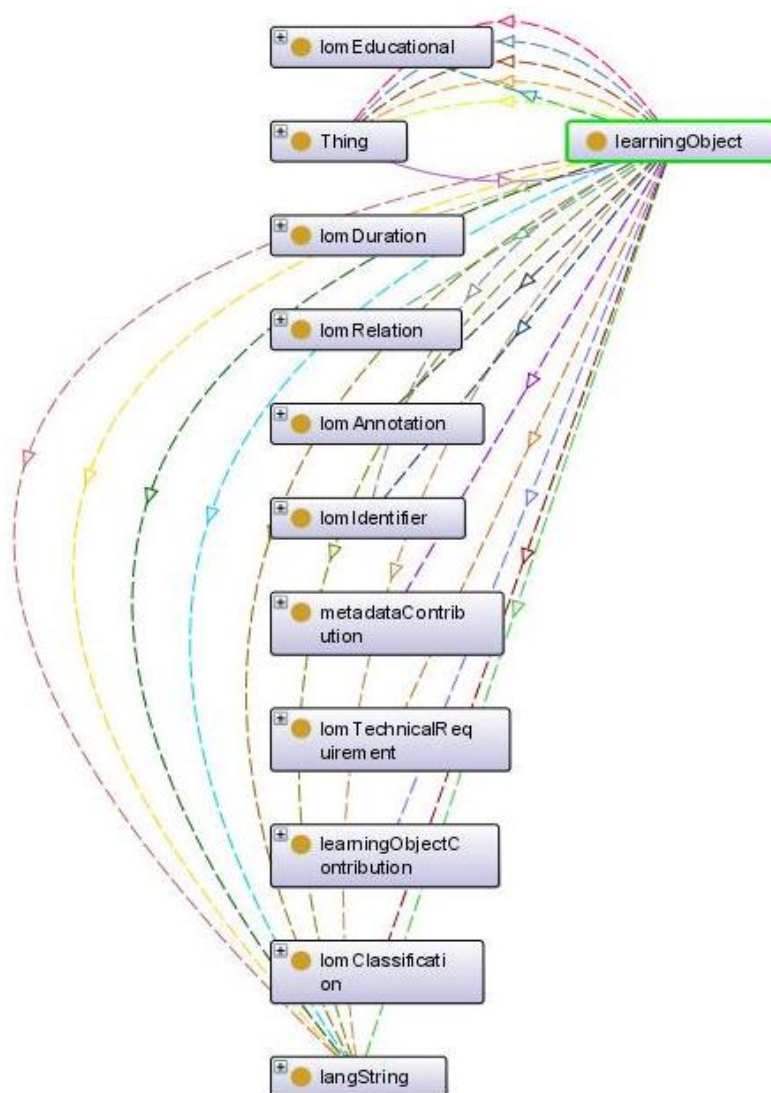


Imagen. 2.16 Relaciones directas de la clase learningObject del estándar LOM. Fuente: Autores

En la tabla 2.5 se presentan las principales clases ontológicas con sus respectivas categorías o subcategorías LOM a las que equivalen y las propiedades de la ontología mediante las cuales se relacionan.

Clase ontología LOM2OWL	Categoría LOM equivalente	Propiedades clase LOM2OWL
contribution learningObjectContribution metadataContribution	La característica “contribución” aparece en dos categorías LOM, por lo que se define una subclase para cada una de ellas: LearningObjectContribution → 2.3. Ciclo de vida.	entityContribution dateTimeContribution



	Contribución. MetadataContribution → 3.2. Metadatos. Contribución	
lomAnnotation	8. Anotación	annotationDescription annotationEntity date_that_this_annotation_was_created
lomClasification	9. Clasificación	clasificationPurpose clasificationTaxonPath taxonDescription
lomTaxonPath	9.2. Clasificación. Ruta Taxonómica	clasificationSource clasificationTaxon
lomTaxon	9.2.2. Clasificación. Ruta Taxonómica. Taxón	taxonEntry taxonId
lomIdentifier	1.1. General. Identificador 3.1. Metadatos. Identificador 7.3. Relación. Identificador	inCatalog isEntry
lomRelation	7. Relación	createdSource relationKind
lomThecnicalRequirement	4. Técnica	
singleTechnicalRequirement	4.4. Técnica. Requisitos	hasType maxVersion minVersion requirementName
vcard		hasEmail hasName hasOrganization
learningObject	Objeto de aprendizaje: Todas las categorías sirven para describir alguna propiedad de un objeto de aprendizaje. Es la clase fundamental de la ontología	Tiene muchas propiedades, una para cada característica o categoría y subcategoría del objeto. Sólo vamos a mencionar algunas de ellas, las que aparecen una sola vez en cada objeto (cardinalidad 1), junto a la categoría LOM a la que

		<p>corresponden:</p> <p>hasAggregationLevel → 1.8. General. Nivel de agregación</p> <p>hasCopyRightRestrictions → 6.2. Derechos. Derechos de autor.</p> <p>hasCost → 6.1. Derechos. Coste.</p> <p>hasDifficulty → 5.8. Uso educativo. Dificultad.</p> <p>hasDuration → 4.7. Técnica. Duration.</p> <p>hasInstallationRemark → 4.5 Técnica. Pautas instalación.</p> <p>hasInteractivityLevel → 5.3. Uso Educativo. Nivel de Interactividad. hasInteractivityType → 5.1. Uso educativo. Tipo interactividad.</p> <p>hasOtherPlatformsRequirement → 4.6. Técnica. Otros requisitos plataforma.</p> <p>hasSemanticDensity → 5.4. Uso educativo. Densidad semántica.</p> <p>hasSize → 4.2. Técnica. Tamaño.</p> <p>hasStatus → 2.2. Ciclo de vida. Estado.</p> <p>has Structure → 1.7. General. Estructura.</p> <p>hasTitle → 1.2. General. Título.</p> <p>hasTypicalLearningTime → 5.9. Uso educativo. Tiempo típico de aprendizaje.</p> <p>hasVersion → 2.1. Ciclo de vida. Versión.</p> <p>hasMetadataLanguage → 3.4. Metadata. Idioma</p>
--	--	--

Tabla 2.5 Principales clases ontológicas con sus respectivas categorías y subcategorías LOM.

### 2.6.2.2 PROV-FOAF

En el documento publicado por la W3C titulado “Dublin Core to PROV mapping” se presenta un mapeo entre metadatos del estándar Dublin Core y clases y propiedades de la ontología PROV, que a su vez puede ser más enriquecida

haciendo uso de la ontología FOAF. («Dublin Core to PROV Mapping», 2013). Esto, debido a que individualmente estas ontologías presentan ciertas limitaciones al momento de describir un OA, pero al fusionarlas permiten una descripción más completa.

**PROV.** La ontología PROV proporciona un conjunto de clases, propiedades y restricciones que se pueden utilizar para representar e intercambiar información sobre la procedencia generada en los diferentes sistemas y en diferentes contextos. Esta también puede ser usada para crear nuevas clases y propiedades a fin de modelar la información.

La ontología PROV define un modelo que permite el intercambio de informaciones que proviene de la Web.

“PROV considera tres elementos básicos: entidades, actividades y agentes. Esos tres elementos están interconectados por un conjunto de relaciones. Por ejemplo, "una entidad (una página Web, un archivo, etc.) fue generada por una actividad asociada a un determinado agente". (Laufer, 2015). La imagen 2.17 presenta una relación básica entre estos tres elementos. Se presenta un diagrama grafico definido por PROV para la representación de ontologías, donde se puede ver que estos tres elementos están relacionados mediante propiedades de esta ontología: `wasDerivedFrom`, `wasAttributeTo`, `wasGeneratedBy`, `wasAssociateWith` y `used`

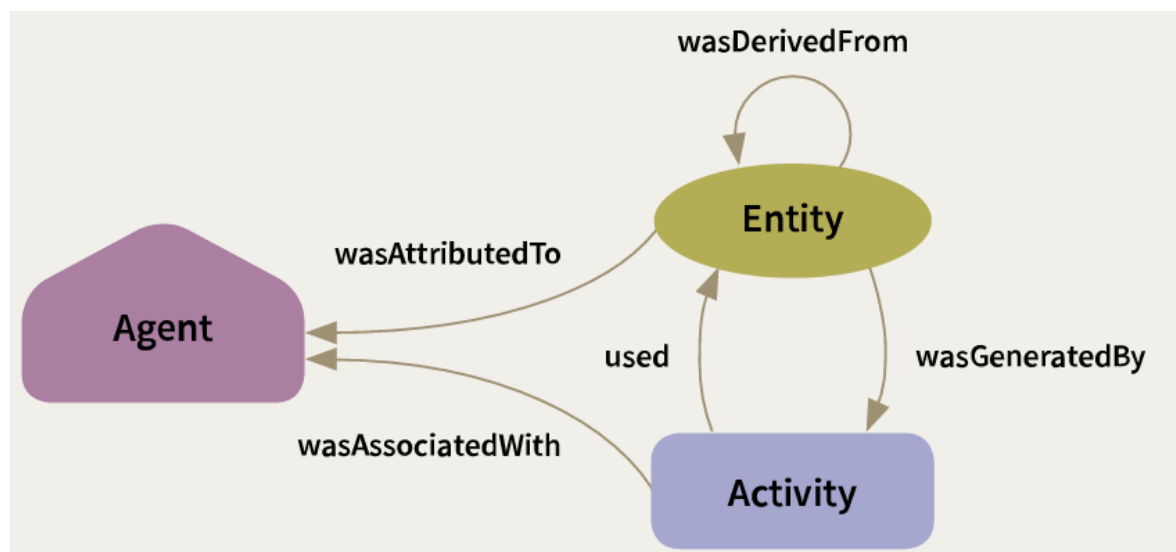


Imagen. 2.17 Entidad, Actividad y Agente (PROV). Fuente <http://ceweb.br/guias/web-semantic/es/capitulo-6/>

**FOAF (Friend of a Friend).** FOAF es una ontología relativamente pequeña, esta cuenta con 19 clases, 44 ObjectProperties y 27 DataProperties.

Consiste en un vocabulario con definiciones que usa el lenguaje RDFS/OWL. FOAF permite describir personas, grupos y documentos mediante la creación de una base de datos unificada de información al normalizar, por ejemplo, categorías y tipos de relaciones. Esta ontología es adecuada para la definición de metadatos sobre personas, sus intereses, sus relaciones y sus actividades. Esta cuenta con un conjunto central de clases y propiedades. Pero además de ese núcleo central, existe una extensión de clases y propiedades relacionadas con las características sociales de la Web, tales como “nick”, “mbox” (e-mail), “homepage”, “publications”, “account”, etc. (Brickley & Miller, 2014)

### 2.6.2.3 LOCO (Learning Object Context Ontologies)

LOCO es un framework que integra una serie de ontologías relacionadas con el aprendizaje. Inicialmente este fue encaminado a facilitar la reutilización de OA y el aprendizaje de diseños. Y luego se extendió también para proporcionar soporte para el aprendizaje personalizado. (Jovanović, Gašević, & Torniai, 2016)

Este framework define la representación formal del contexto del OA y sus principales elementos básicos:

- Los diversos tipos de actividades relacionadas con el aprendizaje,
- Los participantes en el proceso de aprendizaje (es decir, los estudiantes, maestros, asistentes de enseñanza, y similares),
- y los contenidos de aprendizaje.

De acuerdo con ello, LOCO contiene las siguientes ontologías:

- **Learning Context Ontology.** Esta es una ontología de contexto de aprendizaje
- **User Model Ontology.** Esta es una ontología para modelar los participantes de un proceso de aprendizaje, es decir, estudiantes, profesores, autores de contenido, etc.
- **Learnign Object Content Structure Ontology.** Identifica formalmente los objetos de información dentro de un OA con el objetivo de que cada componente sea accesible y reutilizable.
- **Quiz ontology.** Esta es una ontología pequeña para la representación formal de los instrumentos de evaluación que tienen la forma de un cuestionario
- **Domain Ontology.** Es una ontología que permite modelar formalmente la materia de los contenidos de aprendizaje;

- **Learning Design Ontology.** Esta ontología está inspirada en el estándar IMS, el cual está dirigido a la representación formal de los bloques de construcción básicos de un diseño instruccional.

#### 2.6.2.4 Ontología del dominio del Sistema Generador de Ambientes de Enseñanza - Aprendizaje Constructivistas Basados en Objetos de Aprendizaje (AMBAR)

Esta ontología fue desarrollada con la finalidad de describir el dominio AMBAR que es un sistema que se centra en proporcionar a los estudiantes y profesores, una plataforma tecnológica que permita el almacenamiento, generación, uso y reutilización de OA en ambientes educativos. La plataforma tecnológica diseñada es compatible con los estándares actuales en el área, tales como SCORM y la especificación IMS Learning Design (IMS) propuesta para la estandarización y descripción de los procesos de enseñanza-aprendizaje. (López, Miguel, & Montaña, 2000)

### 2.7 Población de Ontologías

El proceso de poblado de una ontología consiste en extraer y representar, de manera manual, semiautomática o automática los individuos, las clases, las relaciones entre los individuos y los valores de las propiedades (Maricela Bravo José A. Reyes-Ortiz, Oscar Herrera, & Alejandro Gudiño, 2015). Todo esto sobre una ontología determinada.

Es posible realizar este proceso de poblado de forma manual, a través de herramientas como Protege; pero de esta manera se vuelve una tarea larga, tediosa y propensa a errores humanos. También, es posible crear procesos de semi automatización y automatización de los mismos. Para el proceso de poblado de una ontología se ha tomado en consideración los estándares mencionados en el punto 2.4.1, teniendo en cuenta principalmente el estándar LOM y el Dublin Core.

LOM es un estándar de metadatos para describir las características o categorías de un OA, y que permiten identificar y clasificar universalmente al mismo. (Fermoso García et al., 2008). O, dicho en otras palabras, este es un estándar específico para OA. Y a Dublin Core, ya que entre los estándares de catalogación o de metadatos más comúnmente utilizados en las bibliotecas digitales se encuentra la Iniciativa de Metadatos Dublin Core, desarrollada para la descripción de un amplio universo de recursos en red ya que su aplicación es de carácter muy general. (López Guzmán,

García Peñalvo, & Pernías Peco, 2005). Además, muchos ROA utilizan este estándar, entre ellos Dspace.

Otra razón importante para considerar estos estándares es que, entre otros, estos también son compatibles con el protocolo OAI-PMH, el cual es usado para la extracción de metadatos de los ROA. Por lo tanto, las razones mencionadas justifican las ontologías que se analizan en esta sección, las cuales son: LOM2OWL y PROV\_FOAF (fusión de la ontología prov y foaf), las mismas que corresponden a los estándares LOM y Dublin Core. La ontología LOM2OWL está desarrollada y descrita en el trabajo titulado “Una ontología en OWL para la representación semántica de objetos de aprendizaje”, donde sus autores detallan el mapeo entre esta ontología y el estándar LOM (Fermoso García et al., 2008). La tabla 2.5 presenta una parte de esta correspondencia entre clases y propiedades de la ontología LOM2OWL y el esquema de categorías que describe un objeto de aprendizaje LOM. La imagen 2.18 presenta la manera jerárquica como están organizadas estas clases. Así mismo, existen una o más propiedades para cada una de estas clases. Una propiedad para cada una de las características que agrupa, las cuales por lo general se nombran como has<nom\_prop>. Finalmente, para cada una de las propiedades de cada una de las clases, se deberá que definir su dominio, rango, objeto a los que se aplica, tipo de valores que puede tomar, de tipo objeto (ObjectProperty) o tipo de datos (dataType). (Fermoso García et al., 2008)

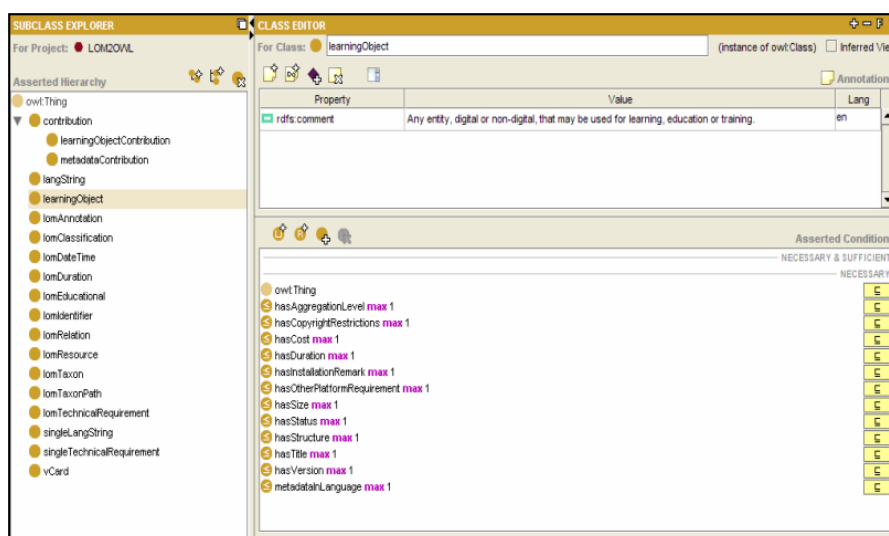


Imagen. 2.18 Jerarquía de clases en LOM2OWL. Fuente:  
[http://www.web.upsa.es/spdece08/contribuciones/176\\_Fermoso\\_Sanchez\\_Sicilia\\_LOMOWL.pdf](http://www.web.upsa.es/spdece08/contribuciones/176_Fermoso_Sanchez_Sicilia_LOMOWL.pdf)

A continuación, se presentan conceptos importantes relacionados con el proceso de poblado de ontologías. (Lendinez Teixeira, 2009)

**Objetos e instancias.** Podrían ser una representación de un objeto en la realidad (por ejemplo, la silla de la habitación). También se conocen como instancias, o individuos.

**Clases.** Representan a un conjunto de objetos. Aunque la palabra “concepto” se utiliza a veces como sinónimo, otros autores relacionan estas dos definiciones diciendo que las “clases” son concretas representaciones de los “conceptos” (ideas de la mente).

**Propiedades.** Que estos objetos pueden tener, siempre dentro del dominio de interés. También se conocen como roles, slots o atributos.

**Relaciones.** Que puede haber entre dichos objetos, que la mayoría de lenguajes ontológicos agrupan dentro de las propiedades de dichos objetos.

### 2.7.1 Poblado Manual

El proceso de poblado manual probablemente es el más importante a tener en cuenta, ya que este permite entender el concepto de creación de individuos, clases, relaciones y propiedades. Este proceso implica utilizar herramientas para el manejo de ontologías, como Protege, con el propósito de crear instancias o individuos, usando las propiedades y las clases para relacionar estas instancias.

En la imagen 2.19 se muestra la interfaz que ofrece Protege, donde se tiene las clases, entidades, propiedades y los individuos de una ontología. Así mismo, esta interfaz permite la creación de instancias o individuos, es decir, permite que un usuario pueda poblar una ontología mediante la creación manual de individuos en dicha ontología.

Todo este proceso es realizado por un usuario o experto que deberá establecer los individuos, relaciones y propiedades manualmente. El desarrollo de esta tarea implica altos costos y consume bastante tiempo al desarrollador de éstas o al experto del dominio. (Gómez Fierros, 2012)

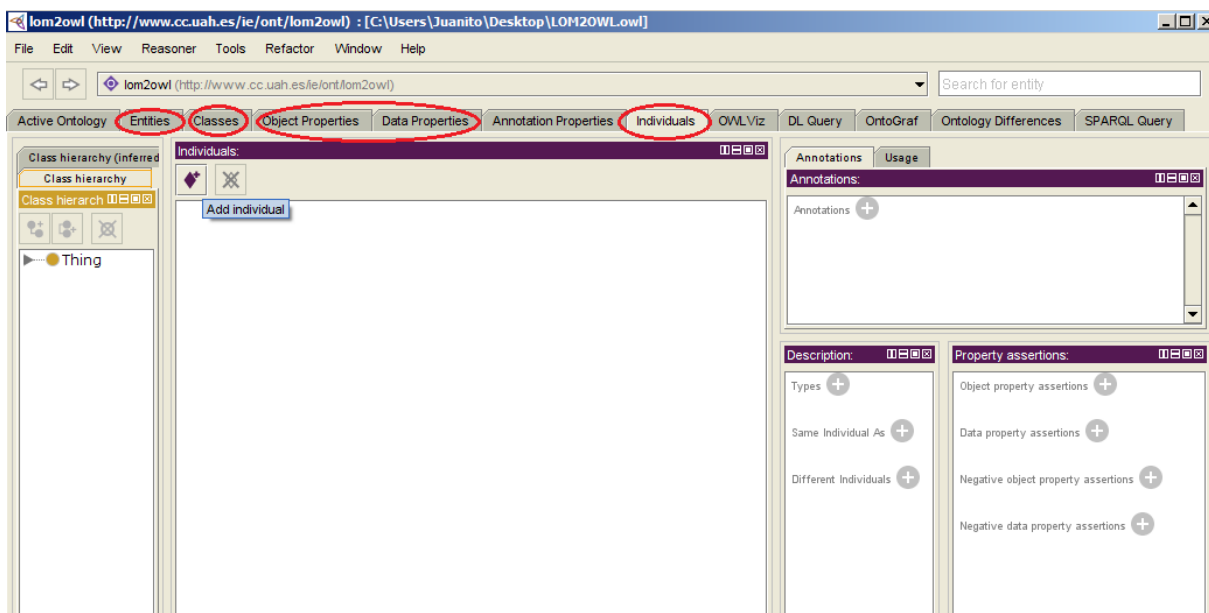


Imagen. 2.19 Interfaz de Protege para el proceso de poblado manual de una ontología

## 2.7.2 Poblado Semiautomático

Este proceso semiautomático consiste en establecer las reglas de mapeo de los metadatos con las clases y propiedades de las ontologías mediante un archivo XML que permita describir estas reglas. Este archivo XML es establecido por un experto (persona), y depende de la ontología con la cual se esté trabajando (Dublin Core, LOM2OWL, etc.). Este proceso de poblado semiautomático consiste en establecer las reglas de entrada al proceso y mediante un algoritmo de programación dar como resultado un archivo RDF poblado con los metadatos del OA. Todo este proceso, evita el arduo trabajo de hacerlo manualmente, el cual se describirá con más detalle en el capítulo 4, al realizarse el respectivo estudio de caso.

## 2.7.3 Poblado Automático

Este proceso de poblado es aquel en el cual no existe la presencia de un experto, es decir, todo es realizado de manera automática. Al no existir un experto que supervise este tipo de poblado, la implementación resulta más compleja que las anteriores.



## 2.8 Herramientas Usadas

### DBpedia

DBpedia es un repositorio en la web, abierto y gratuito, con información estructurada proveniente de Wikipedia. A diferencia de Wikipedia, DBpedia es un conjunto de datos estructurados. Por este motivo, se puede hacer preguntas complicadas a DBpedia que no se podría responder con la búsqueda textual de Wikipedia. (Asociación DBpedia, 2014). El proyecto DBpedia ha generado información semántica a partir de la Wikipedia inglesa. Desde junio de 2011 el proceso de generación de información extrae información de Wikipedia en 15 de sus versiones (idiomas), uno de ellos es el español. En el caso de es.Dbpedia.org, el proceso de extracción produce 100 millones de triples RDF a partir de la versión para el español de la Wikipedia.

### DBpedia Spotlight

DBpedia Spotlight es un sistema que permite el enlace de documentos web con recursos disponibles en DBpedia los cuales son mencionados en el texto. El principal objetivo de DBpedia Spotlight es proporcionar URIs relacionadas a recursos mencionados en un texto, para lo cual realiza un proceso exhaustivo de búsqueda y depuración de resultados (DBpedia Spotlight: Shedding Light on the Web of Documents). (Daiber, Jakob, Hokamp, & Mendes, 2013). DBpedia Spotlight ofrece cuatro funciones básicas para sus usuarios las cuales pueden ser usadas mediante el consumo de servicios web o a través de las API disponible para desarrollo en Java y Scala:

- **Spotting:** Toma un texto de entrada y reconoce entidades o conceptos para anotar. Esto lo realiza mediante varias técnicas de etiquetado como búsquedas en diccionarios o reconocimiento de nombre entidades.
- **Disambiguate:** Toma el texto marcado resultante del proceso anterior donde entidades y conceptos fueron reconocidas y elige un identificador para cada una.
- **Annotate:** Realiza los dos procesos mencionados anteriormente, es decir, toma un texto de entrada y retorna el texto marcado con las URI de las entidades y conceptos identificados.
- **Candidates:** Realiza un proceso similar a annotate, pero este retorna una lista calificada de entidades y conceptos candidatos. Esta lista cuenta con un

grupo de propiedades que describe la calificación de cada candidato por medio de puntajes dados con relación a cuan destacada es la entidad en Wikipedia, por ejemplo, o que tan similar es el significado de la entidad con el texto procesado.

## Dspace

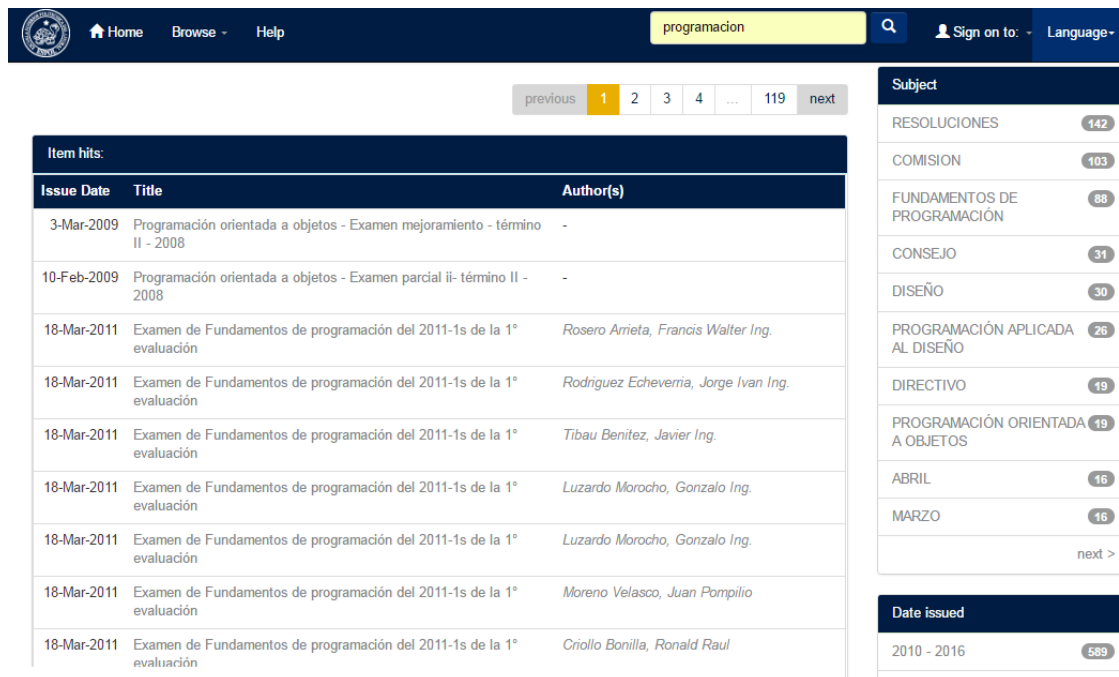
Es una plataforma que ha sido diseñada para ser ejecutada en entornos Web. La misión de esta plataforma es elaborar, registrar y almacenar todo tipo de contenidos digitales. Además, sirve de base para la creación de repositorios institucionales en organizaciones de distinta índole, tales como universidades o centros de investigación alrededor de todo el mundo. Es un sistema de información que captura, almacena, ordena, preserva y distribuye material de investigación digital con el propósito de garantizar que se preserve y distribuya toda la producción intelectual generado al interior de las instituciones que hacen uso de este. (Rodríguez & Sulé, 2008). Está escrito en java, usa una base de datos relacional y soporta el uso de PostgreSQL y Oracle. Utiliza el estándar de metadatos Dublin Core y es compatible con el protocolo OAI-PMH. Se considera a Dspace como un proyecto en continuo crecimiento y que tiene una documentación muy completa. Permite el registro de múltiples tipos de contenidos como tesis, libros, imágenes, videos, audios, etc. A cada objeto digital almacenado se lo denomina ítem, y cada ítem está almacenado en al menos una Colección, la cual está contenida dentro de alguna Comunidad. (Cajal et al., 2012) Un dato importante respecto a este repositorio es que el mismo está configurado con el esquema de metadatos Dublin Core

## Comunidades

Una comunidad pueden ser entidades administrativas tales como escuelas, departamentos, laboratorios y centros de investigación. Dentro de cada comunidad puede existir un número ilimitado de sub-comunidades y de colecciones. Cada colección puede contener un número ilimitado de artículos.

## Colecciones

Las colecciones pueden estar organizadas de acuerdo a un tema, un tipo de información (tales como trabajos o conjuntos de datos) o cualquier otro método de organización que la comunidad considere útil para organizar sus artículos digitales.



The screenshot shows the DSpace interface with a search bar containing 'programacion'. Below the search bar, there are navigation links: 'previous', '1', '2', '3', '4', '...', '119', and 'next'. The main content area displays a table of search results with columns 'Issue Date', 'Title', and 'Author(s)'. To the right of the table, there are two sidebars: 'Subject' and 'Date issued'. The 'Subject' sidebar lists various subjects with their respective counts, and the 'Date issued' sidebar lists dates with their counts.

Issue Date	Title	Author(s)
3-Mar-2009	Programación orientada a objetos - Examen mejoramiento - término II - 2008	-
10-Feb-2009	Programación orientada a objetos - Examen parcial ii- término II - 2008	-
18-Mar-2011	Examen de Fundamentos de programación del 2011-1s de la 1ª evaluación	Rosero Arieta, Francis Walter Ing.
18-Mar-2011	Examen de Fundamentos de programación del 2011-1s de la 1ª evaluación	Rodriguez Echeverria, Jorge Ivan Ing.
18-Mar-2011	Examen de Fundamentos de programación del 2011-1s de la 1ª evaluación	Tibau Benitez, Javier Ing.
18-Mar-2011	Examen de Fundamentos de programación del 2011-1s de la 1ª evaluación	Luzardo Morocho, Gonzalo Ing.
18-Mar-2011	Examen de Fundamentos de programación del 2011-1s de la 1ª evaluación	Luzardo Morocho, Gonzalo Ing.
18-Mar-2011	Examen de Fundamentos de programación del 2011-1s de la 1ª evaluación	Moreno Velasco, Juan Pompilio
18-Mar-2011	Examen de Fundamentos de programación del 2011-1s de la 1ª evaluación	Criollo Bonilla, Ronald Raul

Imagen. 2.20 Dspace de la ESPOL (Escuela Superior Politécnica del Litoral). Búsqueda de recursos referentes a programación.

Es importante señalar algunos términos importantes respecto a esta herramienta:

## Ítems

Cada ítem representa un parte de una colección y el conjunto de ítems representa una colección dentro de una comunidad

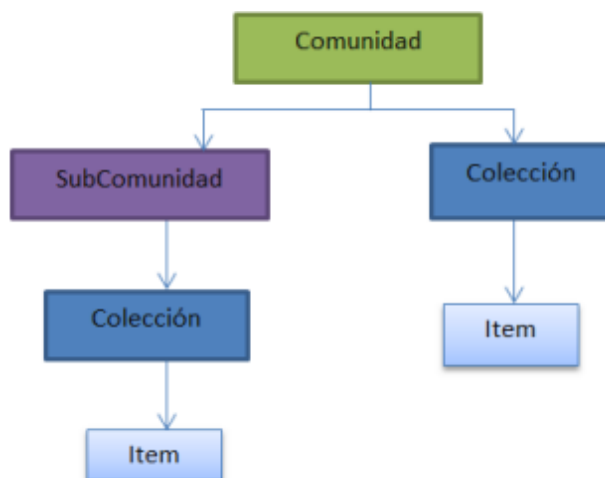


Imagen. 2.21 Comunidad, Colección, ítem de Dspace

## Protege

Protege es un entorno para edición de ontologías creado por la Universidad de Stanford que soporta diferentes lenguajes. Cuenta con un entorno de trabajo que puede ser personalizado por el usuario, el cual permite crear y editar ontologías; incluye herramientas gráficas, las cuales nos permite navegar y editar relaciones de las ontologías fácilmente; incorpora razonadores lógicos que permiten determinar inconsistencias dentro de las ontologías y un endpoint para realizar consultas SPARQL. Además, entre sus funciones principales destacan la opción de fusión de ontologías, renombrado de entidades en otros.

## Apache Jena

Es un framework gratis y de código abierto que nos permite construir aplicaciones de Web Semántica y Linked Data. Este framework cuenta con diferentes APIs que trabajan juntas para procesar, leer y escribir datos RDF, RDFS y OWL. Cuenta con un motor de inferencia que razona sobre ontologías RDF y OWL, permite crear almacenar triples RDF ya sea en memoria o en ficheros, además dispone de un motor de consultas compatible con especificaciones SPARQL y GRDDL. (w3c, 2013)

## Apache Marmotta

Apache Marmota es una plataforma abierta para Linked Data creada con el objetivo de proporcionar una plataforma que pueda ser usada, extendida y desplegada fácilmente por organizaciones que desean desarrollar y publicar aplicaciones de Linked Data. Marmotta permite la lectura y escritura de datos a través de negociaciones de contenido HTTP, permite crear bases de datos de ontologías (RDF triple store), generar y además cuenta con un API para el desarrollo de aplicaciones Linked Data (Sesame). («Apache Marmotta», 2013)

## Netbeans IDE

Es un entorno de desarrollo de código abierto y gratuito para aplicaciones en lenguaje java principalmente el cual permite crear aplicaciones multiplataforma ya que soporta sistemas operativos Linux, Mac y Windows. Actualmente es el IDE

oficial de java 8, cuenta con editores y analizadores de código fuente los cuales permiten crear tanto aplicaciones de escritorio como aplicaciones web. Netbeans además cuenta con una gran cantidad de módulos que extenderlo y aumentar su funcionalidad.

## MySQL

MySQL es un sistema para la gestión de bases de datos relacionales (RDBMS) de código abierto creado por Oracle Corporation, bajo la licencia dual PLP/Licencia Comercial. MySQL está basado en el lenguaje de consultas SQL y funciona sobre plataformas Linux, Unix y Windows.

## Sparql

Es un lenguaje que permite realizar consultas sobre almacenes de datos en forma de triples RDF. (Murua Olalde & Urbano, 2013)

SPARQL define cuatro tipos de consultas (ECSDI, 2015):

- **SELECT:** Permite obtener una lista de tripletas que coinciden con unas propiedades
- **ASK:** Permite saber si existe alguna instancia que cumpla unas propiedades
- **DESCRIBE:** Retorna algunas propiedades del URI que corresponde a la consulta
- **CONSTRUCT:** Permite construir un grafo RDF a partir de los resultados de la consulta

SPARQL define cuatro tipos de operaciones de modificación (ECSDI, 2015):

- **INSERT DATA:** Permite insertar nuevas tripletas
- **INSERT:** Permite insertar nuevas tripletas o mover tripletas de un grafo RDF a otro
- **DELETE DATA:** Permite borrar tripletas
- **DELETE:** Permite borrar tripletas según un patrón

Una consulta SELECT está compuesta por (ECSDI, 2015):

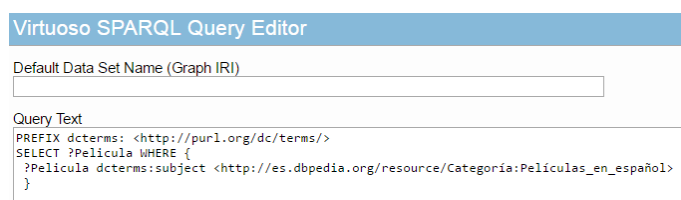
- **BASE:** URI de la TS sobre la que hacer la consulta

- PREFIX: Lista de prefijos a usar para simplificar la referencia a los espacios de nombre
- SELECT: Descripción del resultado de la consulta (qué variables se usarán)
- FROM: Grafo en el que hacer la consulta
- WHERE: Patrón de la consulta
- Modificadores como ORDER BY, GROUP BY, LIMIT, OFFSET, ...

## Ejemplos de consultas SPARQL en DBpedia

Las siguientes consultas se han realizado desde el endpoint de DBpedia en castellano:

### Ejemplo 1.- Películas en español



Virtuoso SPARQL Query Editor

Default Data Set Name (Graph IRI)

Query Text

```
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT ?Pelicula WHERE {
  ?Pelicula dcterms:subject <http://es.dbpedia.org/resource/Categoría:Películas_en_español> .
}
```

Imagen. 2.22 Ejemplo 1. Listado de películas en español

Pelicula
<a href="http://es.dbpedia.org/resource/Cleopatra_era_Cándida">http://es.dbpedia.org/resource/Cleopatra_era_Cándida</a>
<a href="http://es.dbpedia.org/resource/Fando_y_Lis">http://es.dbpedia.org/resource/Fando_y_Lis</a>
<a href="http://es.dbpedia.org/resource/Fiebre_de_amor_(película)">http://es.dbpedia.org/resource/Fiebre_de_amor_(película)</a>
<a href="http://es.dbpedia.org/resource/Plácido">http://es.dbpedia.org/resource/Plácido</a>
<a href="http://es.dbpedia.org/resource/Traffic_(película)">http://es.dbpedia.org/resource/Traffic_(película)</a>
<a href="http://es.dbpedia.org/resource/Amor_a_la_española">http://es.dbpedia.org/resource/Amor_a_la_española</a>
<a href="http://es.dbpedia.org/resource/Domingo_de_carnaval">http://es.dbpedia.org/resource/Domingo_de_carnaval</a>
<a href="http://es.dbpedia.org/resource/Faustino_Mayta_visita_a_su_prima">http://es.dbpedia.org/resource/Faustino_Mayta_visita_a_su_prima</a>
<a href="http://es.dbpedia.org/resource/La_cruel_Martina">http://es.dbpedia.org/resource/La_cruel_Martina</a>
<a href="http://es.dbpedia.org/resource/La_patota">http://es.dbpedia.org/resource/La_patota</a>

Imagen. 2.23 Resultado de consulta SPARQL del ejemplo 1

### Ejemplo 2.- Director de una película

**Virtuoso SPARQL Query Editor**

Default Data Set Name (Graph IRI)

Query Text  

```
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
SELECT ?Nombre WHERE {
  <http://es.dbpedia.org/resource/Solas_(película)> dbpedia-owl:director ?Director .
  ?Director dbpedia-owl:abstract ?Nombre .
}
```

Imagen. 2.24 Ejemplo 2. Director de una película

Nombre
"Benito Zambrano (n. Lebrija, 20 de marzo de 1965) es un guionista y director de cine español."@es

Imagen. 2.25 Resultado de consulta SPARQL del ejemplo 2

## 2.9 Conclusiones del capítulo

Actualmente existe una gran cantidad de OA en Internet. Así mismo existen almacenes o repositorios de información que pretenden gestionar de manera eficiente estos recursos. Estos repositorios contienen OA los cuales a su vez contienen metadatos que permiten describirlos bajo un estándar general como Dublin Core o uno específico para OA como LOM.

Entre los ROA existentes, está presente el repositorio Ágora, el cual cuenta con una vasta cantidad de OA, descritos con sus respectivos metadatos. Además, cuenta con un buscador que permite hacer consultas de OA en base a un criterio dado. Adicionalmente existen herramientas como Dspace que es una plataforma que permite gestionar recursos digitales bajo el estándar Dublin Core, y básicamente sirve para la creación y el manejo de repositorios institucionales en organizaciones de diferente índole, como universidades y centros de investigación.

El proceso de poblado de ontologías consiste en la extracción de metadatos y de representarlos en una ontología determinada tales como LOM2OWL o PROV\_FOAF las cuales trabajan sobre estándares específicos de metadatos. Este proceso se puede realizarlo de manera manual, semiautomática o automática. Para los procesos semiautomáticos o automáticos es necesario contar con un



mecanismo de extracción de metadatos que facilite la obtención de los mismos como es el caso del protocolo OAI-PMH el cual permite el acceso a los metadatos provenientes de distintas fuentes, plataformas y repositorios de contenidos, los mismos que devuelven metadatos en formato XML para su utilización mediante este protocolo.



## Capítulo 3. Diseño de la solución

### 3.1 Introducción

En este capítulo se describe el diseño de la solución propuesta para la anotación de OA mediante tecnologías semánticas y la semi automatización del proceso de poblado de ontologías con metadatos de OA. Para empezar, se describe las funciones que realiza la solución propuesta denominada Metadata Harvester, continuando con la descripción de cada una de las funciones incluyendo las herramientas utilizadas para su implementación. El proceso parte con la extracción de metadatos a través del protocolo de transmisión de contenido OAI-PMH, seguido por la especificación de reglas de mapeo entre la estructura de los metadatos recuperados y las ontologías usadas en la población, luego se describe los pasos seguidos en la población basada en las reglas de mapeo establecidas y la vinculación con recursos externos de DBpedia, y finalmente almacena la ontología resultante en un servidor Apache Marmotta.

### 3.2 Descripción de la solución propuesta

La solución propuesta, denominada Metadata Harvester consiste en varios módulos como se muestra en la imagen 3.1, desarrollados con la finalidad de contar con una herramienta que permita:

- Trabajar con repositorios que implementen el protocolo de transmisión de contenido OAI-PMH utilizado en el presente trabajo para la extracción de metadatos de objetos de aprendizaje.
- Establecer reglas de mapeo entre los metadatos recuperados en formato XML y una o varias ontologías, las cuales servirán para la representación ontológica de los OA a través de sus metadatos.
- Realizar el proceso de poblado de ontologías de uno o varios OA a partir del archivo XML que establece las reglas de mapeo.
- Procesar el texto correspondiente a las palabras clave (*keywords*) o tema (*subject*) del OA a través de la herramienta DBPedia Spotlight con la finalidad de vincular los OA con una fuente externa (DBPedia).
- Almacenar la ontología resultante del proceso descrito en un servidor de Apache Marmotta de la cual se podrá consultar sus datos.

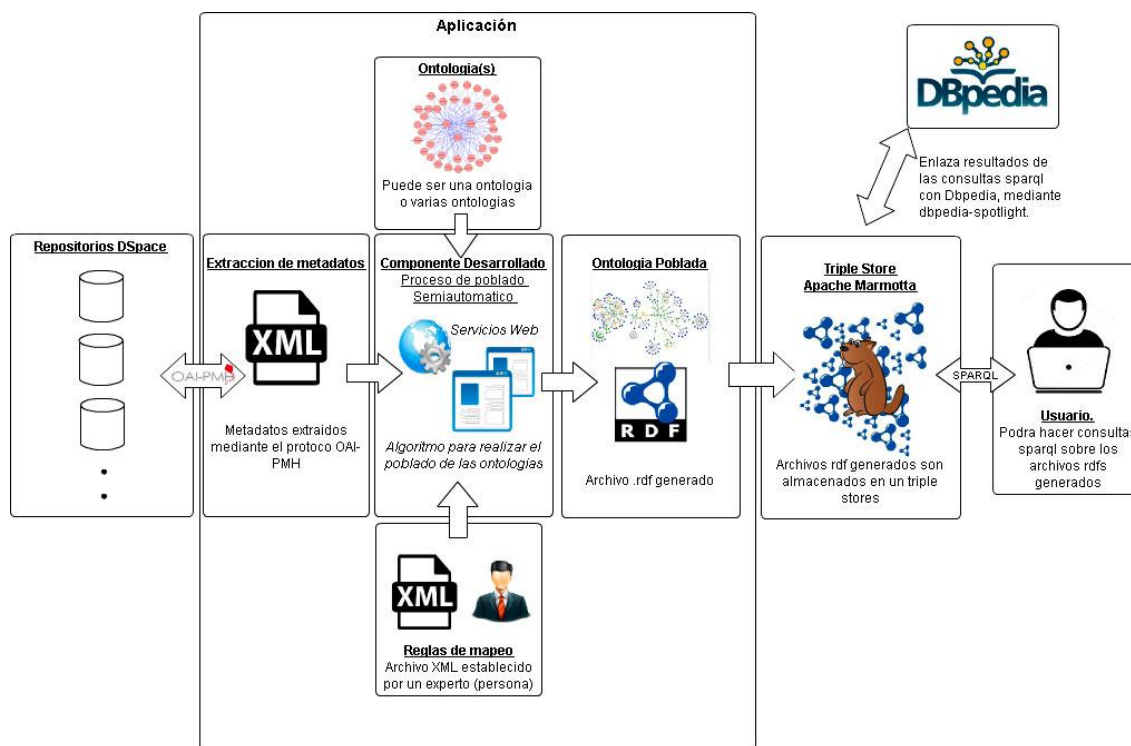


Imagen. 3.1 Descripción de la solución propuesta. Fuente: Autores

La solución propuesta permite realizar la anotación semántica de AO agilizando así el proceso de poblado de ontologías y reduciendo el tiempo que toma realizarlo. Esta solución está basada en el poblado semiautomático descrito en secciones anteriores. Como se indicó la aplicación desarrollada se comunica a través de protocolo OAI-PMH con cualquier repositorio digital, para el presente proyecto se ha implementado un repositorio Dspace como repositorio fuente por ser el más ampliamente utilizado según Opendoar (OpenDOAR, 2016).

El módulo central creado consiste en una aplicación web, la cual cuenta con una base de datos MySQL para almacenar información de los distintos repositorios con los que se podría trabajar, y la implementación de varios servicios web que realizan el trabajo de lectura de las reglas de mapeo establecidas en la anotación de metadatos y de poblado de ontologías. En las secciones posteriores se describirá a detalle el proceso que realiza el sistema implementado.

### 3.3 Entorno de trabajo de la aplicación

La aplicación desarrollada cuenta con tres opciones dentro de su menú principal como se muestra en la Imagen 3.2. A continuación se presenta la función de cada una de las opciones.

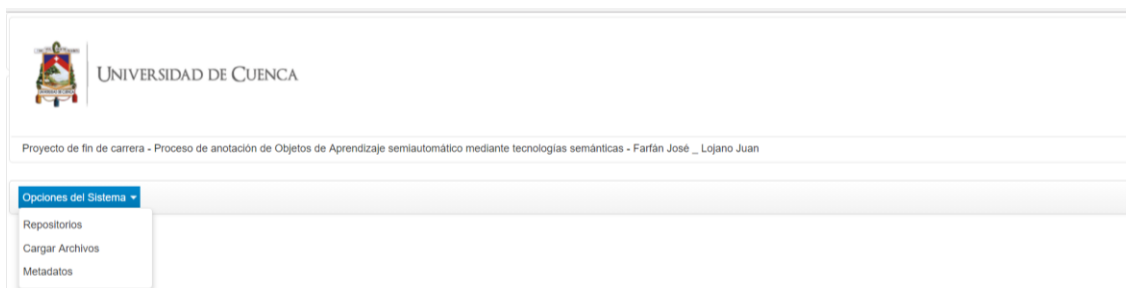


Imagen. 3.2 Menú principal Metadata Harvester. Fuente: Autores

**Repositorios:** La opción repositorios permite realizar el mantenimiento de la información de repositorios en la base de datos; crear, editar y eliminar registros (Imagen 3.3). Estos repositorios deben cumplir con los requerimientos OAI-PMH para su correcto funcionamiento, es de suma importancia el registro correcto de la URL base del repositorio, ya que a través de esta se generará las peticiones de contenido mediante HTTP para la extracción de metadatos.



Imagen. 3.3 Interfaz para el mantenimiento de información de repositorios. Fuente: Autores.

**Cargar Archivos:** Esta opción permite cargar los archivos de configuración para el proceso de poblado. Como se muestra en la Imagen 3.4 los tipos de archivos que deben ser subidos al servidor, el archivo de configuración XML el cual especifica las reglas de mapeo entre los metadatos y las ontologías y las ontologías que puede ser una o varias de acuerdo a lo especificado en el archivo de configuración.

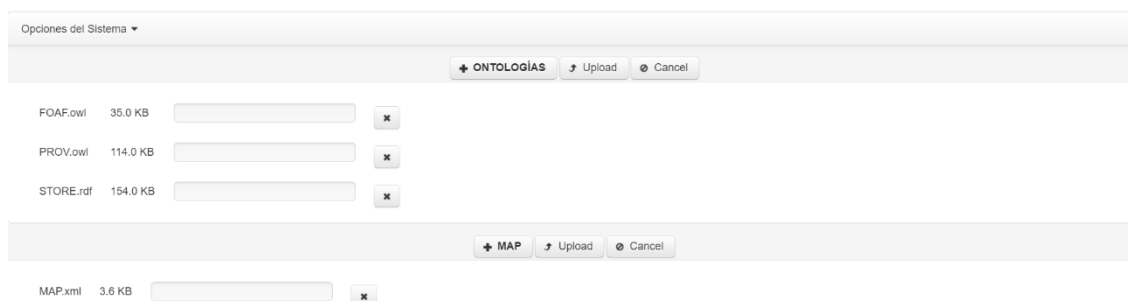


Imagen. 3.4 Interfaz para cargar archivos de configuración. Fuente: Autores.

**Metadatos:** Dentro de esta opción es posible la extracción de los metadatos de los distintos repositorios y el poblado de las ontologías, para lo cual previamente de

debe registrar el repositorio en el sistema y establecer las reglas de mapeo y las ontologías a usar en el proceso de poblado. Todo el proceso consta de cuatro etapas, partiendo en la primera (Repositorio) con una lista donde se puede seleccionar uno de los repositorios registrados, dentro de la segunda etapa (Grupos) el sistema presentará los grupos o colecciones de OA encontrados en el repositorio, en la tercera etapa (Objetos) se presentaran todos los OA que corresponden a la colección seleccionada, teniendo la opción de seleccionar uno o varios OA para realizar el poblado de la ontología, finalmente en la cuarta etapa (Metadatos) se presentan resultados del proceso de anotación realizado a través de DBPedia Spotlight sobre la ontología poblada. En las secciones posteriores se detalla el proceso de interacción con el repositorio digital para la extracción de metadatos, el poblado de ontologías y la anotación mencionados en este apartado



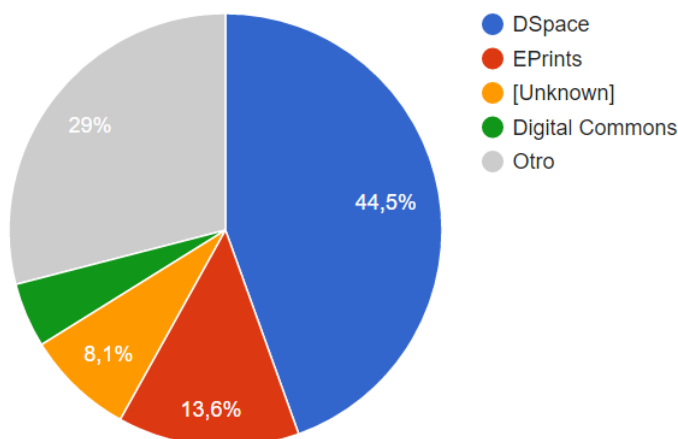
Imagen. 3.5 Interfaz para el proceso de poblado de ontologías. Fuente: Autores.

### 3.4 Repositorios de OA

La solución desarrollada utiliza el protocolo OAI-PMH como medio de comunicación para la obtención de metadatos de distintos repositorios, siendo este el canal de comunicación y las fuentes de OA cualquier repositorio adaptado este protocolo. Luego de analizar algunas alternativas se eligió a Dspace como repositorio en el ambiente de pruebas por ser el más utilizado según la medición realizada por OpenDOAR en el año 2016 (OpenDOAR, 2016) imagen 3.2, dentro de los repositorios que cumplen con dicho protocolo; además de contar de una amplia documentación en comparación de otras opciones analizadas como ARIADNE. Para la instalación y configuración de Dspace se debe cumplir los prerequisites y el proceso mencionados en los anexos del presente trabajo.

Como se observó en la imagen 3.1 los repositorios forman el punto de partida de la solución desarrollada, ya que corresponden a la fuente de los metadatos a usar en la anotación semántica. Para comunicarse con un repositorio es indispensable contar con la URL base sobre la cual incluyen los parámetros para las solicitudes y así obtener los metadatos de los OA.

Usage of Open Access Repository Software - Worldwide



Total = 3334 repositories

OpenDOAR - 06-Mar-2017

Imagen 3.2 Uso de repositorios digitales. Fuente: OpemDOAR

### 3.5 Extracción de metadatos a través del protocolo OAI-PMH



Imagen. 3.2 Comunicación entre repositorios y Metadata Harvester. Fuente: Autores

El segundo componente involucrado en la aplicación desarrollada (imagen 3.1) es el encargado de la comunicación con los repositorios a través de peticiones bajo el protocolo OAI-PMH. La interacción entre un repositorio y el sistema se resume en la imagen 3.2. Cada repositorio cuenta con un servidor que pone a disposición un grupo de servicios web, los cuales son consumidos por Metadata Harvester y devuelven la información solicitada en formato XML.

Repositorio Grupos Objetos Metadatos

Imagen. 3.3 Etapas dentro del proceso de extracción de metadatos. Fuente: Autores

Como se mencionó anteriormente, Metadata Harvester realiza la extracción de metadatos de distintos repositorios por medio de peticiones HTTP bajo el estándar OAI-PMH. En la sección 3.3 se presentó el entorno de trabajo de la aplicación, en esta sección se describe el proceso de comunicación con los repositorios y el procesamiento de información recibida de los mismos. En la transición de cada una de las etapas mostradas en la imagen 3.3 y mencionadas anteriormente intervienen las siguientes peticiones bajo el estándar OAI-PMH.

```
URL_BASE_SELECCIONADA + ?verb=ListSets
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/ http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2016-12-04T17:08:23Z</responseDate>
  <request verb="ListSets">http://memory.loc.gov/cgi-bin/oai2_0</request>
  <listSet>
    <set>
      <setSpec>coll</setSpec>
      <setName>
        Collections of historical content with digitized items
      </setName>
      <setDescription>
        <oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/ http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
          <dc:title xml:lang="en">
            Collections of historical content with items digitized by, or in partnership with, the Library of Congress
          </dc:title>
          <dc:creator>Library of Congress</dc:creator>
          <dc:description>
            Records for collections of digital items mounted for public access at the Library of Congress. Collections are available online through American Memory [http://memory.loc.gov/], Global Gateway [http://international.loc.gov/intldl/intldlhome.html], and elsewhere on the Library of Congress web site. See http://www.loc.gov/library/libarch-digital.html.
          </dc:description>
          <dc:subject xml:lang="en">History--Sources</dc:subject>
          <dc:coverage xml:lang="en">United States</dc:coverage>
          <dc:coverage xml:lang="en">Russia</dc:coverage>
          <dc:type xml:lang="en">image</dc:type>
          <dc:description xml:lang="en">
            Set characteristics for coll: Source records are MARC from LC catalog [catalog.loc.gov]; MODS or oai_dc records are dynamically generated using generic transformation when harvested. dct:accrualPolicy: Passive. dct:accrualPeriodicity: Completely irregular. Contains about 150 records as of March 2008.
          </dc:description>
        </oai_dc:dc>
      </setDescription>
    </set>
  </listSet>
</OAI-PMH>
```

Imagen. 3.4 Petición de lista de colecciones en un repositorio OAI-PMH. Fuente: Autores

La primera petición es la solicitud de la lista de colecciones existentes por medio de la petición ListSets. En la parte superior de la imagen 3.4 encontramos la estructura de la petición realizada, para lo cual se debe concatenar la URL base del repositorio registrada en la base de datos local y previamente seleccionada en la lista de repositorios con el tipo de petición a realizar. En la parte inferior de la imagen se presenta un fragmento XML de la respuesta obtenida, esta respuesta es procesada con la finalidad de obtener el identificador (*setSpec*) y el nombre (*setName*) para ser presentado en pantalla y continuar con el proceso de filtrado de información.



URL\_BASE\_SELECCIONADA + ?verb=ListRecords&amp;metadataPrefix=oai\_dc&amp;set=" + ID\_COLECCION\_SELECCIONADA

```
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/ http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2016-12-04T17:12:38Z</responseDate>
  <request verb="ListRecords" metadataPrefix="oai_dc" set="coll">http://memory.loc.gov/cgi-bin/oai2_0</request>
  <ListRecords>
    <record>
      <header>
        <identifier>oai:lcoai1.loc.gov:lccn/00528551</identifier>
        <timestamp>2010-12-29T15:56:33Z</timestamp>
        <setSpec>coll</setSpec>
      </header>
      <metadata>
        <oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/ http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
          <dc:title>
            African-American sheet music, 1850-1920 selected from the collection of Brown University.
          </dc:title>
          <dc:creator>
            Library of Congress. National Digital Library Program.
          </dc:creator>
          .
          .
          .
        </oai_dc:dc>
      </metadata>
    </record>
  </ListRecords>
</OAI-PMH>
```

Imagen. 3.5 Petición de lista de OA en una colección de un repositorio OAI-PMH. Fuente: Autores.

La segunda petición que se realiza es la solicitud de la lista de OA en la colección seleccionada mediante una petición *ListRecords*. Como se presenta en la imagen 3.5 la petición se obtiene al concatenar la URL base del repositorio seleccionado, los parámetros para la petición de records, el verbo *metadataPrefix* con el valor *oai\_dc*, el cual indica que retorne únicamente metadatos Dublin Core, y el identificador de la colección seleccionado en el paso anterior. Esta petición devuelve una lista completa con todos los recursos incluidos en el grupo como se presenta en el fragmento de la imagen 3.5. La respuesta es procesada para obtener una lista de identificadores y los títulos de todos los OA de la colección, los cuales son presentados en la aplicación.

URL\_BASE\_SELECCIONADA + "?verb=GetRecord&amp;metadataPrefix=oai\_dc&amp;identifier=" + ID\_OA\_SELECCIONADO

```
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/ http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2016-12-04T17:16:47Z</responseDate>
  <request verb="GetRecord" identifier="oai:lcoai1.loc.gov:lccn/00528589" metadataPrefix="oai_dc">http://memory.loc.gov/cgi-bin/oai2_0</request>
  <GetRecord>
    <record>
      <header>
        <identifier>oai:lcoai1.loc.gov:lccn/00528589</identifier>
        <timestamp>2008-05-01T15:49:49Z</timestamp>
        <setSpec>coll</setSpec>
      </header>
      <metadata>
        <oai_dc:dc xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/ http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
          <dc:title>
            Omaha Indian music from the American Folklife Center, Library of Congress.
          </dc:title>
          <dc:creator>American Folklife Center.</dc:creator>
          <dc:subject>Omaha Indians--Music--History and criticism.</dc:subject>
          <dc:description>
            Ethnographic field collection of traditional Omaha music from the 1890s through the 1980s. Contains recordings from wax cylinders, songs, speeches, interviews, photographs, field notes, and tape logs. Features material from the 1983 Omaha harvest celebration pow-wow, 1985 Hethu'shka Society concert at the Library of Congress, and 1983 and 1999 interviews with members of the Omaha tribe.
          </dc:description>
          <dc:publisher>[Washington, D.C.] : Library of Congress</dc:publisher>
          <dc:date>2000</dc:date>
          <dc:identifier>http://hdl.loc.gov/loc/afcc/colfafe.af000007</dc:identifier>
          <dc:language>eng</dc:language>
        </oai_dc:dc>
      </metadata>
    </record>
  </GetRecord>
</OAI-PMH>
```

Imagen. 3.6 Petición de metadatos de un OA en un repositorio OAI-PMH. Fuente: Autores.

La tercera y última petición es la solicitud de los metadatos del OA seleccionado, esta es una petición *GetRecord*, la cual devuelve los metadatos de un OA seleccionado a través de su identificador. La imagen 3.6 muestra la estructura de la petición, la cual contiene la URL base del repositorio seleccionado concatenado con los parámetros para la petición *GetRecord* y el identificador del OA seleccionado, adicionalmente y al igual que en la petición mencionada anteriormente se especifica que se desea obtener los metadatos Dublin Core mediante el verbo *metadataPrefix*. La respuesta XML obtenida contiene todos los metadatos del OA los cuales serán procesados para posteriormente realizar el poblado de la ontología de acuerdo a un grupo de reglas obtenidas del proceso de anotación.

### 3.6 Ontologías

Metadata Harvester consiste, como se mencionó al iniciar este capítulo, de una aplicación web la cual interactúa con los repositorios de OA a través del protocolo OAI-PMH, el servidor encargado del procesamiento de ontologías y el usuario que proporciona las entradas para realizar el procesamiento de información. En este apartado se describe cómo se especifican las ontologías que intervienen en la anotación de OA.

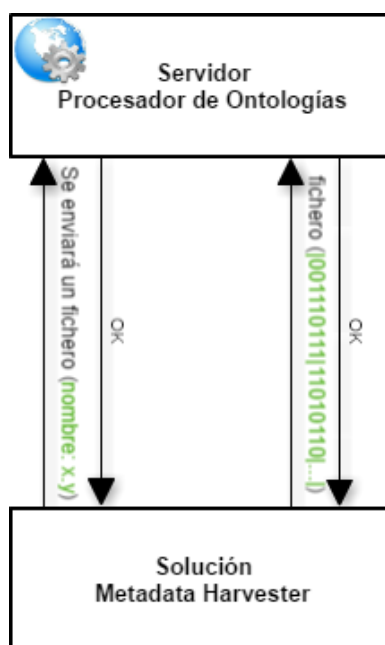


Imagen. 3.7 Envío de ficheros al servidor. Fuente: Autores.

Entre los servicios web con los que trabaja el sistema tenemos dos destinados a la carga de ficheros en el servidor. La imagen 3.4 presentó la interfaz la cual permite enviar archivos al servidor, ahora en la imagen 3.7 se presenta el proceso seguido



para el envío. En primera instancia se envía un mensaje al servidor indicando que a continuación se enviará un fichero, en este mensaje se especifica el nombre del fichero ya que este el nombre es importante para identificarlos en el procesamiento posterior. Una vez anticipado el envío del fichero y recibido el mensaje de confirmación desde el servidor se procede a enviar el fichero por medio de un segundo servicio que recibe un flujo de bytes que forman el fichero en el servidor en un directorio destinado para los mismos.

Metadata Harvester permite cargar tantas ontologías como necesite dentro de la anotación. La carga de ficheros descrita anteriormente pone a disposición dentro del servidor los archivos RDF u OWL que en lo posterior serán leídos y procesados con el framework de Apache Jena.

### 3.7 Mapeo manual y generación de reglas de mapeo

Hasta el momento se han mencionado dos de las tres entradas con las que trabaja Metadata Harvester; los metadatos extraídos de los repositorios de OA, las ontologías representadas en ficheros RDF u OWL; a continuación, se mencionará la entrada principal analizada en el presente trabajo que consiste en el archivo de reglas de mapeo resultado de la anotación realizada.

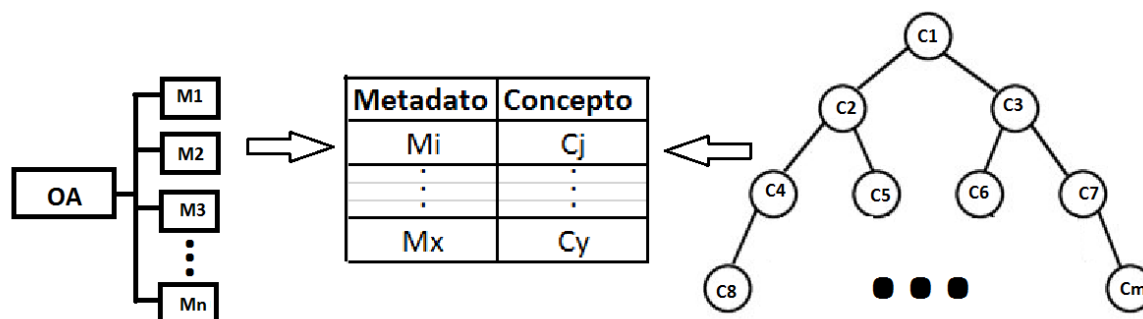


Imagen. 3.8 Anotación de OA. Fuente: Autores.

Uno de los factores clave para el correcto poblado de ontologías a través de la aplicación Metadata Harvester radica en el proceso de anotación analizado en este apartado, ya que este dará como resultado el archivo de configuración que contiene las reglas de mapeo entre metadatos y ontologías. Previa la generación de reglas de mapeo es indispensable que el usuario realice el proceso de poblado manual de la o las ontologías que intervendrán, esto con la finalidad de identificar la representación ontológica de los metadatos.

El objetivo de la anotación es establecer una relación entre distintos metadatos de un OA con conceptos de ontologías semánticas (imagen 3.8). La relación que se plantea indica que un metadato representa la misma información que un concepto en una ontología, esto sugiere que no todos los metadatos se podrían relacionar con un concepto de una ontología y viceversa, esto dependerá de la finalidad para la que fue creada la ontología. Por ejemplo, si una ontología fue creada para representar específicamente un formato de metadatos lo más probable es que todos los metadatos y conceptos intervengan en la anotación, pero, por otro lado, si se trabaja con una ontología de propósito general o que abarca un ámbito más grande que los metadatos de un OA existirán concepto o incluso metadatos que no intervendrán en la anotación.

Cabe recordar que la aplicación permite trabajar con varios formatos de estándares de metadatos representados en formato XML y varias ontologías, debiendo realizarse la identificación de reglas de mapeo para cada combinación entre estándar de metadatos y ontologías a utilizar.

### 3.7.1 Identificación de reglas de mapeo

Mediante el poblado manual el usuario será capaz de identificar la estructura que tiene los metadatos de un OA en términos de una o más ontologías. Al analizar a detalle la estructura de los metadatos de OA y el significado que tiene cada uno de ellos, al igual que la estructura y el significado de cada concepto de una ontología se puede realizar la anotación semántica considerando lo siguiente. Para empezar, debe existir una clase en una de las ontologías la cual se representa un OA y dos tipos de relación básicas:

- El primer tipo de relación es el que se identifica entre uno de los metadatos y el OA directamente, es decir, será necesario únicamente un triple para su representación en términos de las ontologías usadas. Por ejemplo, el metadato *dc:title* el cual contiene el título dentro del estándar Dublin Core se podría vincular directamente con la representación del OA a través de la propiedad *foaf: title* de la ontología FOAF, en estas relaciones interviene únicamente un *data property* una clase y el metadato como tal.
- El otro tipo de relación es una relación compuesta entre el OA y el metadato, para cuya representación es necesaria la creación de más de un triple, es decir el valor del metadato se vinculará indirectamente con el OA. Por ejemplo, el metadato *dc: publisher*, el cual contiene el nombre del publicador, se vincula con la clase *prov: Agent* a través de la propiedad *foaf: givenName*,

y este a su vez se vincula con la representación del OA a través de la propiedad *prov: publisher*. Es decir que en este caso para la representación del metadato interviene un *data property*, uno o más *object property* junto con más de una clase.

### 3.7.2 Representación XML de las reglas de mapeo

El resultado del proceso de anotación se lo representa en un archivo XML para su posterior lectura y procesado por parte de la aplicación y así realizar el proceso de poblado de ontologías. Para dicha representación se utiliza los siguientes prefijos dentro de las etiquetas XML para identificar los elementos de las ontologías a usar:

- **cl\_NombreOntología\_**: Los primeros dos caracteres indica que se hará referencia a una clase dentro de en la ontología que lleva el nombre especificado, este nombre especificado en el prefijo debe ser el mismo con el que es cargado la ontología al servidor como se indicó en el apartado 3.6.
- **op\_NombreOntología\_**: Para el segundo tipo de prefijo los dos primeros caracteres indican que se hará referencia a una propiedad tipo *object property* dentro de en la ontología que lleva el nombre especificado.
- **dp\_NombreOntología\_**: Finalmente para el tercer tipo de prefijo los dos primeros caracteres indican que se hará referencia a una propiedad tipo *data property* dentro de en la ontología que lleva el nombre especificado.

La imagen 3.9 presenta la representación XML de las reglas de mapeo de un metadato en formato Dublin Core donde interviene dos clases de la ontología, dos propiedades, una de tipo objeto y una de tipo dato y el metadato como tal. El archivo XML tiene como raíz la clase la cual representa el OA (Para el ejemplo la clase *Entity*), a medida que se recorre el texto XML se encuentra la propiedad con la que se relacionará la clase seguida de la clase o valor que representará el predicado del triple a crear. Es así que por ejemplo la imagen 3.9 representaría los siguientes triples:

- **Entity—>Creator—>Agent**: Relación entre un OA (*Entity*) con su creador (*Agent*) con la propiedad *prov: creator*.
- **Agent—>givenName—>dc: creator**: Relaciona al creador del OA (*Agent*) con su nombre, el cual es extraído de los metadatos proporcionados en formato XML resultado de la solicitud http (OAI-PMH).

Como se observa en la imagen la clase *Agent* se encuentra al mismo nivel que la raíz *OAI-PMH* la cual contiene el metadato a procesar (*dc:creator*), esto significa que se creará un individuo (*Agent*) por cada metadato de este tipo en el OA.

```
<cl_PROV_Entity>
  <op_PROV_creator>
    <cl_PROV_Agent>
      <dp_FOAF_givenName>
      </dp_FOAF_givenName>
    </cl_PROV_Agent>
  <OAI-PMH>
    <GetRecord>
      <record>
        <metadata>
          <dc>
            dc:creator
          </dc>
        </metadata>
      </record>
    </GetRecord>
  </OAI-PMH>
</op_PROV_creator>
</cl_PROV_Entity>
```

Imagen. 3.9 Representación XML de una regla de mapeo. Fuente: Autores.

## 3.8 Proceso de poblado de ontologías

El proceso de poblado de ontologías consiste en dos etapas principalmente: La generación de triples a partir de las reglas de mapeo y el proceso de Linked Data, en el cual se vincula cada OA con recursos en DBpedia mediante el uso de DBpedia Spotlight. Hasta ahora se describió cual es el proceso para la generación de las tres entradas que necesita Metadata Harvester para realizar el poblado de ontologías (imagen 3.1), los metadatos de los OA, las ontologías que intervienen en el poblado y las reglas de mapeo resultado del proceso de anotación. A continuación, se describe el proceso de poblado de ontologías a partir de las tres entradas mencionadas.

### 3.8.1 Proceso semiautomático de poblado de ontologías

La población de ontologías parte con los metadatos de los OA en formato XML obtenidos mediante el proceso descrito en la sección 3.5. Una vez obtenidas las tres entradas para el poblado de la ontología se solicita la creación de los triples correspondientes como se muestra en la imagen 3.10. Metadata Harvester brinda la opción de seleccionar uno o más OA de la colección seleccionada previamente como muestra la imagen 3.11. Una vez cargados los archivos al servidor, ontologías y reglas de mapeo, se incluye dentro los parámetros a enviar en la solicitud de poblado la lista de ontologías a utilizar y la lista de OA seleccionados, los cuales incluyen la referencia para el acceso de sus metadatos en el repositorio. Los

parámetros son transformados a formato json e incluidos en la petición de poblado para luego ser procesados por el servidor. Una vez terminado el poblado el servidor responderá con los resultados obtenidos durante el proceso, también en formato json, para ser mostrados en pantalla.

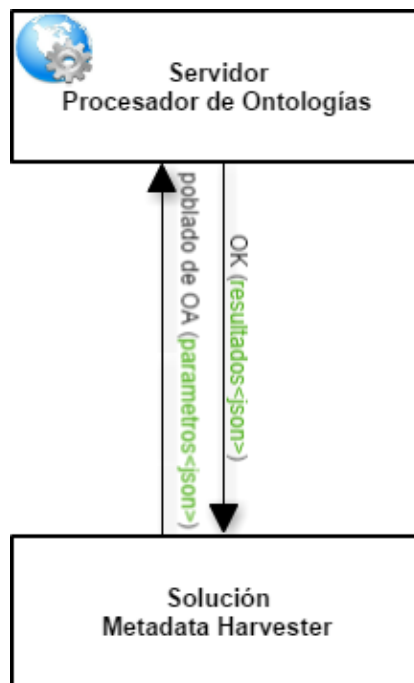


Imagen. 3.10 Envío de parámetros para el poblado de ontologías. Fuente: Autores.

Como un paso preliminar es necesario leer los parámetros recibidos, dentro de los cuales se incluirán la lista de ontología y la lista de OA que intervendrán. En primera instancia se cargan las ontologías en un mapa donde se almacenará el nombre de cada una con su ontología como tal usando el framework de Apache Jena. Luego se leerá el archivo XML que contiene las reglas de mapeo el mismo que se debe cargar previamente al servidor. Finalmente, se recorrerá la lista de OA, cargando para cada uno sus metadatos y realizando la creación de triples.

ID Objeto	Nombre Objeto
<input checked="" type="checkbox"/> oai:icoa1.loc.gov:iccn/00528551	African-American sheet music, 1850-1920 selected from the collection of Brown University.
<input checked="" type="checkbox"/> oai:icoa1.loc.gov:iccn/00528589	Omaha Indian music from the American Folklife Center, Library of Congress.
<input checked="" type="checkbox"/> oai:icoa1.loc.gov:iccn/00528679	Florida Folklife from the WPA Collections 1937-1942
<input checked="" type="checkbox"/> oai:icoa1.loc.gov:iccn/00528680	Prairie settlement Nebraska photographs and family letters, 1862-1912.
<input checked="" type="checkbox"/> oai:icoa1.loc.gov:iccn/00529295	Puerto Rico at the dawn of the modern age nineteenth- and early-twentieth-century perspectives ; from selected divisions of the Library of Congress.
<input type="checkbox"/> oai:icoa1.loc.gov:iccn/00529303	The nineteenth century in print the making of America in books and periodicals.

Imagen. 3.11 Selección de OA previo el proceso de poblado semiautomático. Fuente: Autores.

El proceso de lectura de reglas de mapeo y poblado de ontologías en el servidor cuenta con tres etapas como se muestra en la imagen 3.12. El archivo que contiene

las reglas de mapeo tiene el formato presentado en la parte izquierda de la imagen, el mismo que se puede interpretar como un árbol que llamamos árbol de mapeo y puede ser procesado recursivamente. Dentro del árbol de mapeo se puede presentar tres casos generales, cada uno de los cuales debe ser procesado de manera diferente.

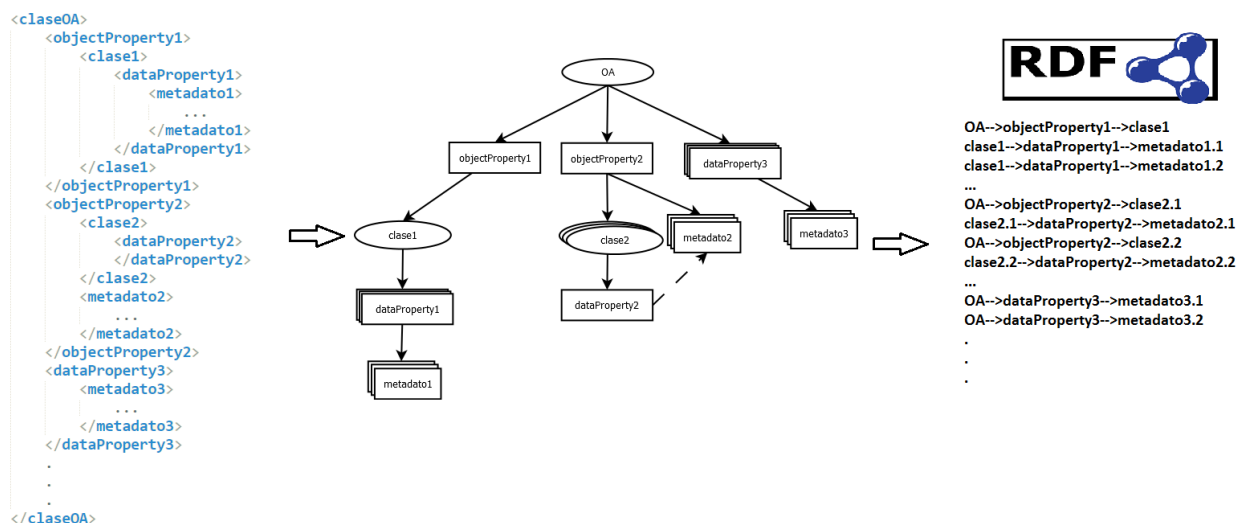


Imagen. 3.12 Lectura de anotación y poblado de ontologías. Fuente: Autores.

El proceso de lectura de reglas de mapeo y poblado de ontologías en el servidor cuenta con tres etapas como se muestra en la imagen 3.12. El archivo que contiene las reglas de mapeo tiene el formato presentado en la parte izquierda de la imagen, el mismo que se puede interpretar como un árbol que llamamos árbol de mapeo y puede ser procesado recursivamente. Dentro del árbol de mapeo se puede presentar tres casos generales, cada uno de los cuales debe ser procesado de manera diferente.

Como se observa en la imagen 3.12 el primer caso parte de la clase de la ontología con la cual se representará al OA, seguida de una o más object properties e igual número de clases, una por cada propiedad y terminando con una propiedad tipo data property y la referencia al metadato que se incluirá en el triple. Para ciertos formatos de metadatos como Dublin Core puede existir más de un valor para un mismo metadato, es por esto que como se observa se crearán tantas instancias de la propiedad como metadatos del tipo metadato1 se hayan recuperado, en la parte derecha de la imagen 3.12 se representa como quedaría los triples resultantes para el procesamiento del primer caso.

Para el segundo caso, como se observa en la imagen, el formato de las reglas de mapeo es bastante similar con la diferencia de la ubicación de la referencia al metadato. Al incluirse la referencia al metadato al mismo nivel de una clase esto indica que se deberán crear tantos individuos de esa clase como metadatos de ese tipo exista, tal como representa la clase2 en la imagen 3.12, el resultado de este procesamiento es varios triples donde interviene la clase2 con todos los triples que se generan luego, finalizando con un data property y el valor del metadato respectivo como se muestra en la parte derecha de la imagen.

El tercer y último caso es el más sencillo de los tres, donde los metadatos se relacionan directamente con el OA a través de un data property. El resultado del procesamiento de reglas de mapeo de este tipo da como resultado tanto triples como metadatos del tipo metadato3 existan. En estos triples intervienen el individuo que representa el OA, la propiedad con la que se están relacionando y los metadatos recuperados.

### 3.8.2 Ontología poblada

Como se observó en el punto anterior, al recorrer el árbol de mapeo la aplicación genera instancias de las clases y propiedades encontradas en el recorrido y genera uno o más triples dependiendo de la cantidad de metadatos que contenga el OA. El proceso descrito anteriormente se repite para cada OA y haciendo uso de Apache Jena son generados los triples descritos en el árbol de mapeo. Posteriormente estos triples son almacenados en un archivo RDF dentro del servidor.

## 3.9 Vinculación de OA con recursos de DBpedia

Una vez realizado el proceso de poblado semiautomático se obtiene como resultado una ontología poblada con los metadatos proporcionados resultado de la extracción del repositorio seleccionado, cuyos triples fueron generados de acuerdo a las reglas de mapeo especificadas en el archivo XML como se indica en el apartado 3.7.2.

Sobre la ontología resultante se realiza una consulta SPARQL con la finalidad de obtener texto descriptivo; ya sea palabras clave, descripciones del contenido del OA u otro texto dependiendo de cada formato de metadatos; de cada OA para el posterior proceso de vinculación con recursos de DBpedia a través de DBpedia Spotlight. En la imagen 3.13 se muestra una consulta que listará los metadatos *subject* extraídos de metadatos bajo el estándar Dublin Core. Esta consulta será única para la ontología sobre la que se trabaje y dependiendo sobre que metadatos se desea trabajar.



```
= "SELECT * WHERE {\n"
+ "  ?object rdf:subject  ?subject\n"
+ "}";
```

Imagen. 3.13 Consulta SPARQL sobre ontología PROV\_FOAF. Fuente: Autores.

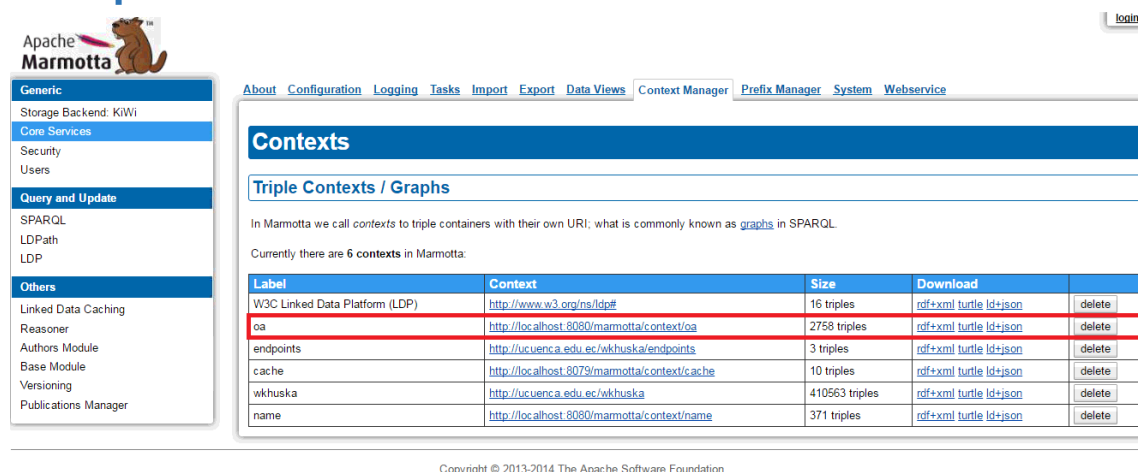
El proceso de anotación mediante DBpedia Spotlight se la realiza a través del consumo de servicios web para lo cual se sigue un proceso similar al utilizado para obtener los metadatos de los repositorios descrito en el apartado 3.5. La imagen 3.14 muestra la estructura de la solicitud http que se realiza, donde se incluye el texto recuperado de cada OA, ya sea este texto palabras clave, descripciones del contenido, entre otros. El texto enviado es procesado por DBpedia Spotlight y devuelve como resultado los recursos relacionados identificados en el proceso de anotación.

[http://spotlight.sztaki.hu:2222/rest/annotate?text= + TEXTO\\_DESCRIPTIVO\\_OA + &confidence=0.2&support=20](http://spotlight.sztaki.hu:2222/rest/annotate?text= + TEXTO_DESCRIPTIVO_OA + &confidence=0.2&support=20)

Imagen. 3.14 Estructura de petición para anotación con DBpedia Spotlight. Fuente: Autores.

Los recursos recuperados con DBpedia Spotlight son vinculados a los OA por medio de la propiedad *rdfs: seeAlso* y almacenados en el archivo RDF, dando como resultado vinculaciones con recursos de DBpedia para cada OA. El resultado de esta vinculación dependerá de la calidad del texto descriptivo de cada OA.

## 3.10 Almacenamiento del archivo RDF resultante en Apache Marmotta



Apache Marmotta

Generic: Storage Backend: KiWi, Core Services: Security, Users, Query and Update: SPARQL, LDPPath, LDP, Others: Linked Data Caching, Reasoner, Authors Module, Base Module, Versioning, Publications Manager.

Contexts

Triple Contexts / Graphs

In Marmotta we call *contexts* to triple containers with their own URI; what is commonly known as *graphs* in SPARQL.

Currently there are 6 contexts in Marmotta:

Label	Context	Size	Download	
W3C Linked Data Platform (LDP)	<a href="http://www.w3.org/ns/ldp#">http://www.w3.org/ns/ldp#</a>	16 triples	<a href="#">rdf+xml</a> <a href="#">turtle</a> <a href="#">ld+json</a>	<a href="#">delete</a>
oa	<a href="http://localhost:8080/marmotta/context/oa">http://localhost:8080/marmotta/context/oa</a>	2758 triples	<a href="#">rdf+xml</a> <a href="#">turtle</a> <a href="#">ld+json</a>	<a href="#">delete</a>
endpoints	<a href="http://ucuenca.edu.ec/wkhushka/endpoints">http://ucuenca.edu.ec/wkhushka/endpoints</a>	3 triples	<a href="#">rdf+xml</a> <a href="#">turtle</a> <a href="#">ld+json</a>	<a href="#">delete</a>
cache	<a href="http://localhost:8079/marmotta/context/cache">http://localhost:8079/marmotta/context/cache</a>	10 triples	<a href="#">rdf+xml</a> <a href="#">turtle</a> <a href="#">ld+json</a>	<a href="#">delete</a>
wkhushka	<a href="http://ucuenca.edu.ec/wkhushka">http://ucuenca.edu.ec/wkhushka</a>	410563 triples	<a href="#">rdf+xml</a> <a href="#">turtle</a> <a href="#">ld+json</a>	<a href="#">delete</a>
name	<a href="http://localhost:8080/marmotta/context/name">http://localhost:8080/marmotta/context/name</a>	371 triples	<a href="#">rdf+xml</a> <a href="#">turtle</a> <a href="#">ld+json</a>	<a href="#">delete</a>

Copyright © 2013-2014 The Apache Software Foundation

Imagen. 3.6 Ontología resultante almacenada en un servidor Apache Marmotta. Fuente: Autores.

El proceso final que realiza Metadata Harvester es el almacenamiento de la ontología poblada en un servidor Apache Marmotta. Apache Marmotta cuenta con un servicio web que permite la publicación de ontologías en su triple store, para lo



cual se especifica un contexto bajo el cual se almacenará la ontología. La imagen 3.15 muestra la ontología publicada en el servidor Apache Marmotta. Una vez almacenada la ontología será posible realizar consultas SPARQL sobre los triples generados a través de la terminal disponible en el mismo servidor.

### 3.11 Recapitulación

Metadata Harvester es la solución propuesta en el presente trabajo de titulación, este permite la comunicación con diferentes repositorios que implementen el protocolo de transmisión de contenido OAI-PMH con la finalidad de obtener metadatos de OA. El objetivo es llevar a cabo un proceso de anotación sobre un formato de metadatos y una o varias ontologías, este proceso consta de análisis de los estándares de metadatos y estructura de las ontologías a utilizar en el establecimiento de reglas de relación o mapeo entre la estructura de los metadatos recuperados y las ontologías usadas en el proceso de población semiautomática.

El sistema consta de una aplicación web la cual requiere de tres estradas: los metadatos recuperados, las ontologías a utilizar y las reglas de mapeo establecidas. Esta aplicación interactúa con los repositorios a través de solicitudes HTTP con la finalidad de recuperar metadatos de OA en formato XML; además el sistema consta de servicios web, los cuales procesan los metadatos recuperados, los representa ontológicamente considerando las reglas de mapeo establecidas y los vincula con recursos externos en DBpedia haciendo uso de la herramienta DBpedia Spotlight. Finalmente publica la ontología resultante en un servidor Apache Marmotta donde el usuario podrá acceder a los metadatos y realizar consultas SPARQL sobre el mismo.

# Capítulo 4. Estudio de Caso.

## Experimento controlado

## Población de Ontologías

### 4.1 Introducción

Luego de haber desarrollado la solución propuesta para el proceso semiautomático de anotación de OA, mediante tecnologías semánticas con consultas SPARQL enlazadas a DBpedia, es importante recoger y analizar la información generada. Por esta razón se ha visto conveniente la realización de un estudio de caso, que consiste en la ejecución de un experimento controlado de población de ontologías con metadatos de OA de un repositorio local de Dspace, donde se evidencia las ventajas de la aplicación “Metadata Harvester”, especialmente en el tiempo de población de las ontologías.

El presente capítulo describe el proceso de realización del estudio de caso, la definición del mismo, ámbitos en los que es relevante el estudio, fuentes de información, objetivos, escenario y ejecución del experimento, los elementos que intervienen, los resultados y su análisis. Es decir, se muestra la ejecución de un estudio de caso de tipo descriptivo de una situación crítica, debido a que el mismo se ajusta al presente experimento, ya que se examina una situación singular de interés único, que sirve para dar una aseveración respecto a los beneficios y ventajas que ofrece aplicación Metadata Harvester.

### 4.2 Estudio de caso

#### 4.2.1 Selección y definición del caso

##### Selección y definición

Se ha seleccionado el estudio de caso de tipo descriptivo, y más específicamente el de tipo descriptivo de una situación crítica, ya que este tipo de estudio de caso examina una situación singular o específica de interés único y puede servir como prueba crítica de una aseveración acerca de un programa, proyecto o estrategia. (Madera Payeta et al., 2013). En este caso se ha decidido emplear la solución desarrollada como producto final del presente proyecto, con el propósito de evidenciar su validez en el proceso de poblado de ontologías frente a un proceso

manual. Y, además crear la vinculación de resultados con recursos de la fuente externa DBpedia.

Básicamente, consiste en un experimento controlado para realizar la población de ontologías, y comparar estos resultados frente al proceso manual. Esto, especialmente en lo que se refiere al número de pasos que implica cada proceso, y por ende el tiempo de duración de los mismos. Además, el experimento también implementa, mediante código java, consultas sobre las ontologías y presenta resultados vinculados con recursos externos de DBpedia.

Se dice que es un experimento controlado debido al escenario en el cual es realizado, en el cual se ha realizado lo siguiente:

- Implementación de un repositorio local de OA, mediante Dspace.
- Se ha procedido a subir un total de 46 OA al repositorio local. Se ha seleccionado este número de OA debido a que es el total de resultados que revuelve el repositorio Ágora al realizar una búsqueda por el criterio: “programación”, como lo muestra la imagen 4.1.



**AGORA**  
Ayuda a la Gestión de Objetos Reutilizables de Aprendizaje

Inicio :: Descripción :: Políticas :: Comentario :: Usuarios conectados

**Listar Recursos**

Para buscar un recurso escribe una palabra o frase que lo identifica, también puedes usar el [visor de metadatos](#) para definir una palabra de búsqueda.

Criterio de búsqueda para OA en Agora

Texto:  [Búsqueda avanzada...](#)

☐ En la red de servidores AGORA (puede ser tardado)

Buscar

Cantidad total de OA devueltos por la consulta en Agora

Recursos listados: 46

Tipo	Recurso	Kb	Publicador	Opciones	Herramientas
Podcast multimedia MP3	<a href="#">Orígenes de Python</a> Introducción a un Lenguaje de Programación usando la metáfora de anuncio de película de cine. Versión para iPod (iTunes) y PC (RealPlayer 11 o QuickTime). Estadísticas: 0	2,977.95	Pacheco Alberto	    	    

Imagen. 4.1 Búsqueda de OA en el repositorio Ágora, por el criterio de “programación”. Cantidad total de OA obtenidos del ROA Ágora (46)

## Ámbitos en los que es relevante el estudio

Es importante estudiar este caso concreto desde el ámbito que ofrece el experimento controlado, dado que el mismo pretende mostrar las ventajas que

proporciona la solución desarrollada para el proceso de poblado semiautomático de ontologías y también pretende mostrar recursos externos de DBpedia a los cuales están vinculados los OA cargados en el repositorio local de Dspace. Es en este ambiente donde se puede desarrollar procesos controlados de población de ontologías y desde el cual se puede extraer la mayor cantidad de información para el estudio de caso.

Por otro lado, los ámbitos en los que es relevante el estudio son en los procesos de poblado de ontologías y vinculación de resultados con fuentes externas, como por ejemplo DBpedia. Otro ámbito importante es el campo de los OA, ya que se pretende poblar ontologías con metadatos de OA y vincularlos con recursos externos, con el objetivo de enriquecer su contenido.

### **Fuentes de información**

La información necesaria para el presente estudio de caso es extraída de los capítulos anteriores (1, 2 y 3), ya que en los mismos se presentan temas importantes que sirven para la formulación del experimento controlado. Por ejemplo, información del proceso de poblado de una ontología, así como de las herramientas necesarias como Dspace, DBpedia, Ágora, Marmotta, etc.

### **Problema**

En este punto, se pretende analizar el problema en el estudio del presente caso, el mismo que consiste en el experimento controlado que contiene los procesos de poblado de una ontología, tanto manual como semiautomática. Por lo tanto, el problema del estudio de caso es el poblado de una ontología con los metadatos extraídos de los OA de un repositorio local. Además, un problema más específico, es el extenso número de pasos que conlleva un proceso de poblado manual, lo cual implica un tiempo excesivo para dicho proceso. Asimismo, el estudio de caso pretende mostrar el proceso de poblado semiautomático mediante la aplicación Metadata Harvester, como un recurso para la reducción de pasos y por ende del tiempo de poblado de una ontología.

### **Objetivos**

Los objetivos del presente estudio de caso son:

- Realizar el proceso de poblado manual de una ontología con los metadatos de un OA obtenidos del ROA Ágora y cargado en el repositorio local de Dspace y registrar el número de pasos que toma dicho proceso y obtener un

tiempo aproximado para el mismo, ya que no es posible obtener un tiempo exacto, pues esto provocaría trabajar con cálculos subjetivos.

- Realizar el proceso de poblado semiautomático de una ontología, mediante la aplicación Metadata Harvester y registrar el número de pasos que toma dicho proceso y también obtener un tiempo exacto para este proceso (en este caso si es posible obtener un tiempo exacto).
- Comparar los resultados de estos procesos, a fin de demostrar la validez de la aplicación Metadata Harvester, bajo el escenario controlado, especialmente en la reducción del número de pasos a realizar para el poblado de una ontología, y por ende la reducción del tiempo de dicho proceso.
- Vincular los OA del repositorio local de Dspace con recursos de la fuente externa DBpedia.

#### 4.2.2 Elaboración de una lista de preguntas

Para este estudio de caso se plantean las siguientes preguntas:

1. ¿Cuál es el proceso de poblado de una ontología?
2. ¿Cuántas maneras existen de poblar una ontología? ¿y cuáles son?
  - a. ¿Cuántos OA se pueden poblar simultáneamente?
3. ¿Cuántos estándares existen para metadatos de OA?
  - a. ¿Es posible poblar ontologías usando cualquier estándar?
4. Al agilizar el proceso de poblado de ontologías ¿Cuánto tiempo se reduciría el mismo?
  - a. ¿Cuánto tiempo se tarda en realizar el poblado de ontologías de forma manual?
  - b. ¿Cuánto tiempo se tarda en realizar el poblado de ontologías usando la aplicación Metadata Harvester?
    - i. ¿Cuántas fuentes externas son vinculadas con los resultados de las consultas a las ontologías?
    - ii. ¿Cuántas tripletas genera el proceso de poblado de la ontología?

c. ¿Cuándo se obtienen los mejores resultados?

### 4.2.3 Localización de las fuentes de datos

La información para el estudio de caso se la ha conseguido de los capítulos 1, 2 y 3. Y la estrategia para la obtención de información que se ha usado para el presente estudio de caso es la investigación, es decir, un estudio de la documentación relacionada con los temas necesarios para el planteamiento del experimento controlado, la misma que se encuentra detallada en los capítulos 2 y 3.

### 4.2.4 Análisis e interpretación

En el presente estudio de caso se va a abordar un análisis del proceso de poblado de una ontología usando metadatos de OA, extraídos de un repositorio local (Dspace), bajo el protocolo Dublin Core. Poblar una ontología consiste en la extracción de metadatos de un OA y la creación de individuos o instancias, clases, propiedades y las respectivas relaciones entre dichas instancias, todo esto bajo un estándar para los metadatos, pudiendo ser estos LOM, Dublin Core, etc. En este sentido el proceso de poblado manual implica un extenso número de pasos, lo cual provoca un tiempo exagerado para este proceso. Es aquí donde la aplicación Metadata Harvester pretende dar solución y mejorar dicho proceso, reduciendo el número de pasos y por ende el tiempo del mismo.

### 4.2.5 Elaboración de un informe

El presente estudio de caso se ha realizado en el escenario que se observa en la imagen 4.1, el mismo que también se describe en el punto 4.2.1

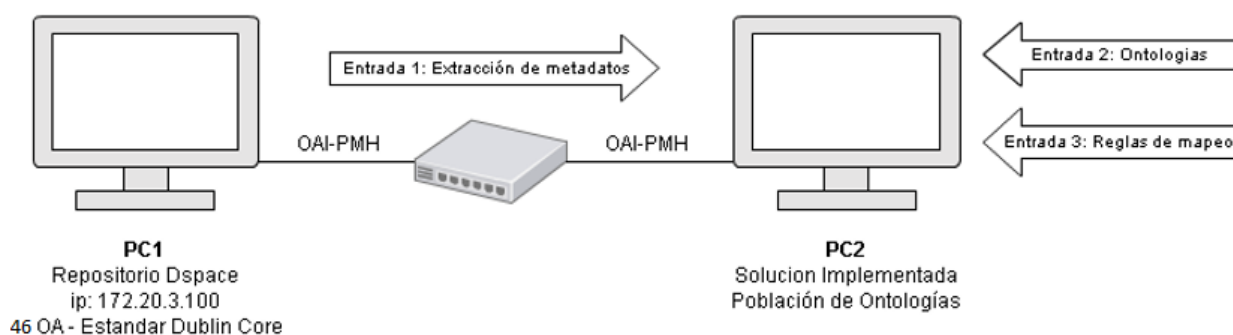


Imagen. 4.2. Escenario en el cual es ejecutado el experimento controlado

Este estudio de caso presenta la aplicación práctica de lo descrito en el capítulo 3, imagen 3.1. A continuación, se muestra la aplicación de cada uno de los módulos de dicha imagen.

## Modulo: Repositorios DSpace

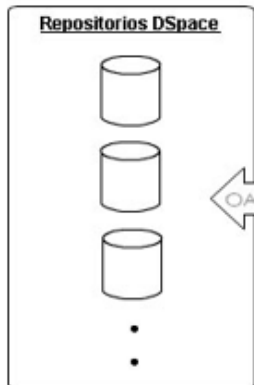


Imagen. 4.3 Módulo de Repositorios - Dspace

Para el presente estudio de caso se ha procedido a realizar la instalación de un repositorio local mediante la herramienta Dspace en una computadora, la misma que servirá como repositorio local de los OA. Este proceso de instalación está incluido en los anexos de este documento. La aplicación práctica del módulo de la imagen 4.3 se puede ver en la imagen 4.4, en donde Metadata Harvester permite seleccionar cualquier repositorio que se encuentre registrado en la misma. En la imagen 4.4 se puede ver el repositorio local, el mismo que contiene los 46 OA. Es importante indicar que Metadata Harvester también permite el registro de nuevos repositorios dentro de su sistema, para lo cual será necesario el nombre del repositorio y una URL válida del mismo, también el nuevo repositorio deberá permitir la comunicación mediante el protocolo OAI-PMH. Por lo tanto, la aplicación tendrá tantos repositorios como se necesiten.

Opciones del Sistema ▼

Repositorio Grupos Objetos Metadatos

REPOSITORIO

LISTA

1 10 ▼

IdRepositorio	Nombre	UriBase
1	Library of Congress	https://memory.loc.gov/cgi-bin/oai2_0
2	Local	http://172.20.3.100:8080/oai/request
4	Preceding	http://precedings.nature.com/oai2

1 10 ▼

Imagen. 4.4 Selección del repositorio Dspace en la aplicación



## Módulo: Extracción de metadatos



Imagen. 4.5 Módulo de extracción de metadatos

Como se indicó en el capítulo 3, la aplicación Metadata Harvester realiza la extracción de metadatos por medio de peticiones HTTP bajo el estándar OAI-PMH. Y el repositorio local de Dspace que contiene los 46 OA para el presente estudio de caso, trabaja bajo este protocolo. De esta manera, la imagen 3.3 muestra cada una de las etapas en la aplicación (Repositorio, Grupos, Objetos) y las peticiones HTTP que se realizan se encuentran detalladas en el capítulo 3, sección 3.5.

En la aplicación Metadata Harvester, la etapa de Repositorios es la que se observa en la imagen 4.4, donde se ha seleccionado el repositorio local de Dspace que se encuentra instalado en la PC1. La siguiente etapa se observa en la imagen 4.6 la misma que presenta los grupos que están dentro del repositorio seleccionado, es aquí donde se encuentra el repositorio como tal que contiene los 46 OA, el cual se llama “Objetos de Aprendizaje”. La explicación entre lo presentado en la imagen 4.4 y 4.6 se encuentra en el capítulo 2, sección 2.8, de donde se puede concluir en que el primero es una comunidad (repositorio), mientras que el segundo es una subcomunidad (repositorio).

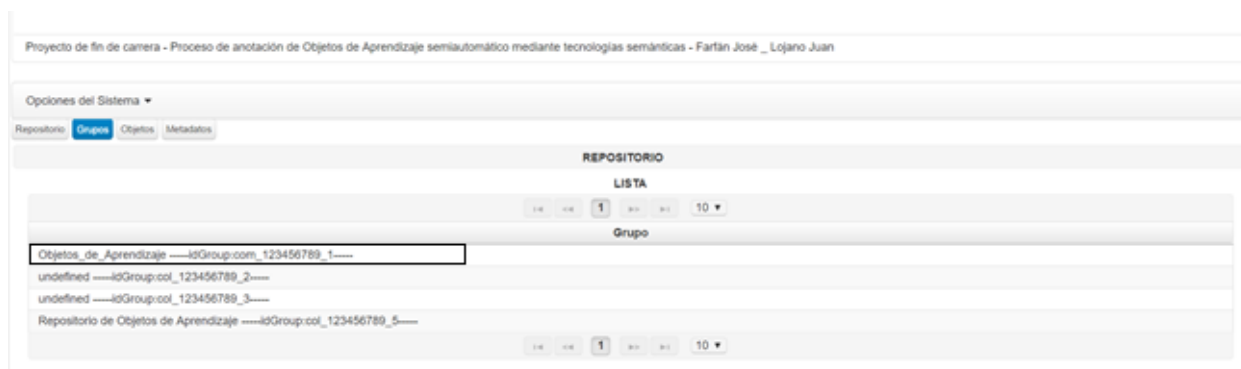


Imagen. 4.6 Elección de grupo de OA: Repositorio de Objetos de Aprendizaje (localizado en el pc1)

La siguiente etapa es la de Objetos, la cual presenta los OA que están contenidos en el repositorio seleccionado, esto se observa en la imagen 4.7.

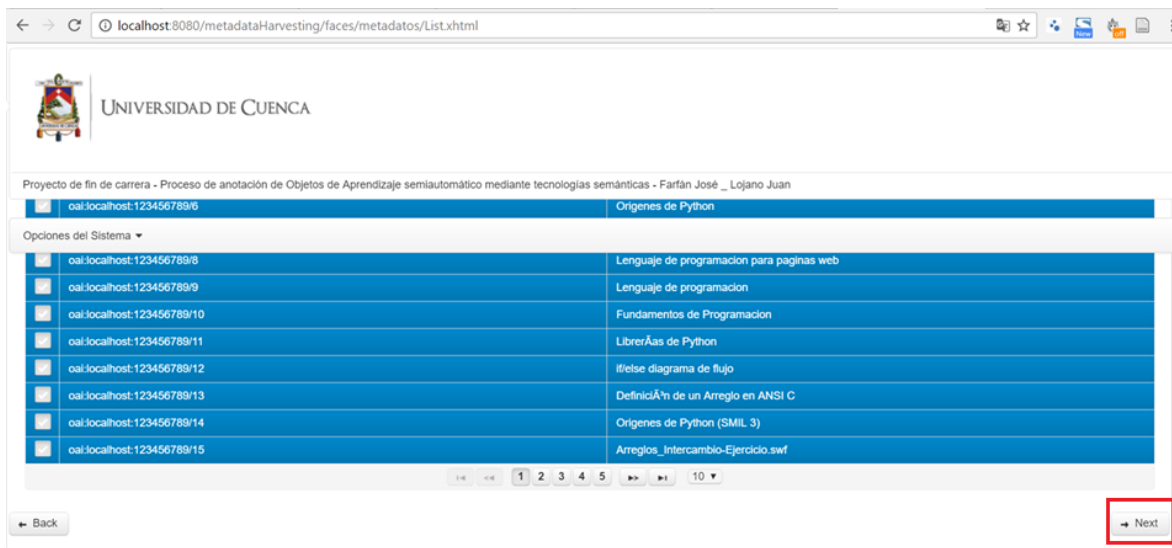


Imagen. 4.7 Selección de todos los OA del repositorio.

Finalmente, se procede al último paso del proceso de este módulo, para el cual se deberá dar clic en el botón Next de la imagen 4.7, lo cual realizará internamente el proceso de extracción de los metadatos de todos los OA seleccionados, además de realizar las demás tareas para el proceso de poblado de la ontología, dando como resultado una ontología poblada con los metadatos de los OA del repositorio local.

## Modulo: Ontologías



Imagen. 4.8 Módulo de Ontologías

Metadata Harvester permite cargar tantas ontologías como se necesiten. El proceso interno de la carga de ficheros se describe en el capítulo 3 en la sección 3.6. En esta sección se describe el proceso de carga de ontologías dentro de la interfaz de la aplicación. En la imagen 3.2 se puede ver el menú principal de Metada Harvester, el cual tiene la opción de “Cargar archivos”, esta opción conduce a la pantalla

presentada en la imagen 4.9, donde se pueden cargar tantos archivos como el proceso de poblado requiera, tanto ontologías como los archivos correspondientes a las reglas de mapeo.

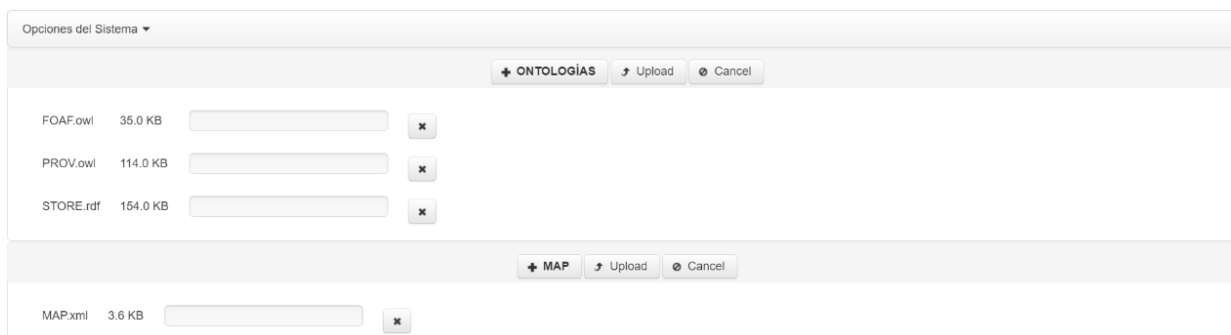


Imagen. 4.9 Cargado de archivos en la aplicación "Metadatos Harvester"

Para el presente estudio de caso se ha procedido a cargar las ontologías FOAF.owl y PROV.owl, además del archivo STORE.rdf que en este caso es necesario ya que las dos primeras ontologías trabajaran juntas. La descripción de cada una de las ontologías cargadas en la aplicación se presenta en el capítulo 2. Y para el presente estudio de caso se usan dos ontologías ya que la una le complementa a la otra.

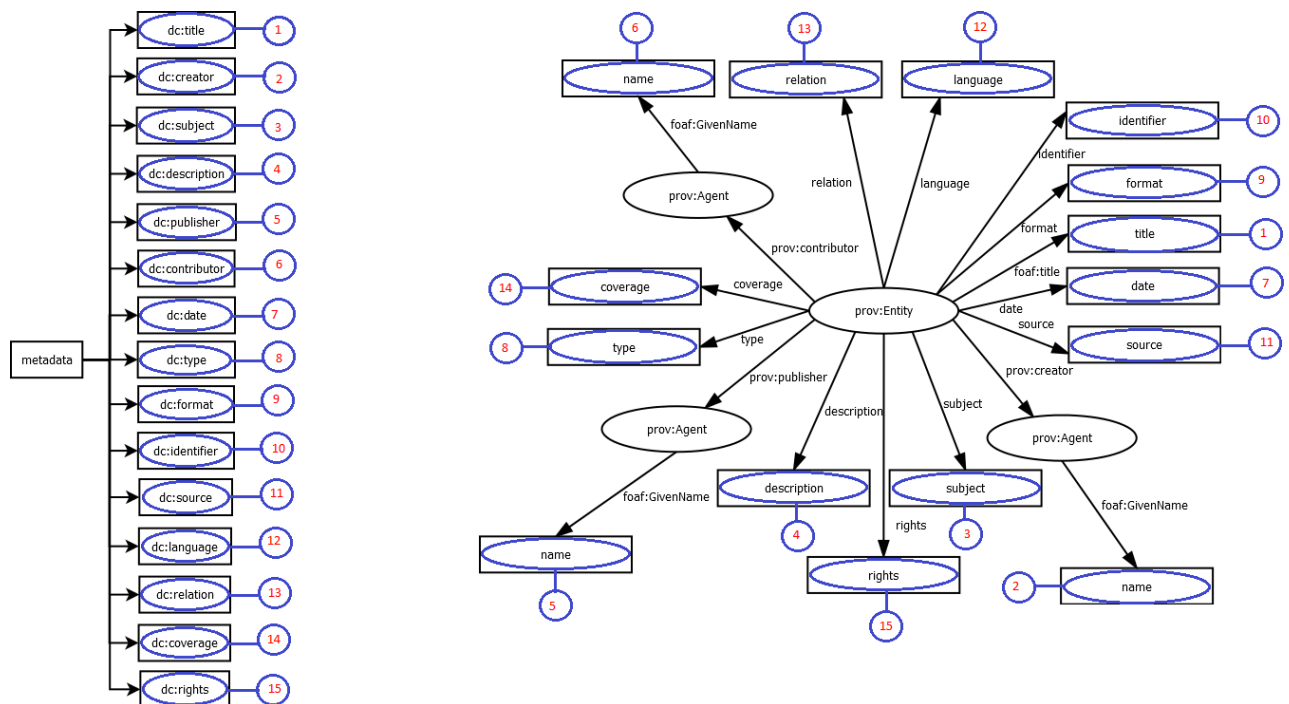
## Módulo: Reglas de mapeo



Imagen. 4.10 Módulo Reglas de mapeo

Para realizar el proceso de poblado de ontologías es necesario representar las reglas de mapeo en formato XML para procesarlas mediante la aplicación. Para dicha representación se ha utilizado los prefijos descritos en el capítulo 3, en la sección 3.7.2. En la imagen 3.9 se puede observar la representación en formato XML de las reglas de mapeo de un metadato en formato Dublin Core. Donde interviene dos clases de la ontología, dos propiedades, y el metadato como tal, que

En este punto es importante indicar que un proceso previo de poblado manual permite identificar la estructura que tendrá los metadatos de un OA en términos de una o más ontologías. La imagen 4.10 presenta la representación gráfica del resultado de identificación de reglas de mapeo para el caso particular entre el estándar PROV y la ontología denominada PROV\_FOAF que es la usada en la aplicación Metadata Harvester. En la imagen se representa tanto la estructura del estándar como de la representación ontológica de los metadatos del OA y se identifica el vínculo que existe entre cada una de las partes. Un proceso de poblado manual de un OA en una ontología se presenta en los anexos del presente documento.



En la aplicación Metadata Harvester, estas reglas de mapeo deben ser cargadas mediante la segunda opción del menú principal, imagen 3.2. La misma que conduce a la pantalla de la imagen 4.9, donde se puede cargar los archivos necesarios con todas las reglas de mapeo para el proceso de poblado. Como nota adicional, se puede mencionar que este archivo XML de reglas es supervisado o realizado por un experto (persona), esta es una característica importante del proceso de poblado semiautomático, el cual es el que se ha implementado en la aplicación Metadata

Harvester. Básicamente, el proceso de poblado semiautomático mediante la aplicación es el siguiente:

**Paso 1.** Se ejecuta la aplicación. La imagen 4.12 muestra la primera pantalla de la misma, donde para comenzar se debe dar clic en el botón “Iniciar”, ubicado en la parte inferior derecha



Imagen. 4.12 Interfaz de la aplicación que implementa la solución. Pantalla de bienvenida

La aplicación cuenta con un menú de “Opciones del Sistema” el cual contiene 3 opciones, como se presenta en la imagen 3.2, donde:

**Repositorios:** Permite realizar el mantenimiento de la información de repositorios en la base de datos; crear, editar y eliminar registros

**Cargar Archivos:** Permite cargar los archivos necesarios para el proceso de poblado, es decir las ontologías y las reglas de mapeo.

**Metadatos:** Permite la extracción de los metadatos del repositorio y el poblado de las ontologías.

**Paso 2.** Mediante la segunda opción del menú: Cargar Archivos, se procede a cargar las ontologías necesarias para el experimento, (en este caso 3 ontologías: FOAF.owl, PROV.owl y STORE.rdf), y también el archivo correspondiente a las reglas de mapeo (MAP.xml). Estos archivos cargados se pueden ver en la imagen 4.9

**Paso 3.** A continuación, se procede a la selección de un repositorio, que para el presente caso será el repositorio local creado en Dspace, del cual se procederá a la extracción de los metadatos de sus OA, esto se muestra en la imagen 4.4.

**Paso 4.** Dentro del repositorio local de Dspace se encuentra la colección creada con el propósito de que la misma actúe como el repositorio local de OA para el presente experimento y el siguiente paso es la elección de la misma, es decir, “Objetos de Aprendizaje”. Esto se puede ver en la imagen 4.6

**Paso 5.** Se debe tener en cuenta que es posible seleccionar uno o más OA. Para el experimento se procedió a seleccionar todos los OA que están en el grupo seleccionado del repositorio local, es decir 46 OA, con el fin de medir el tiempo de poblado en la ontología seleccionada. Esto se puede ver en la imagen 4.7.

**Paso 6.** Finalmente se procede al último paso de este proceso de poblado, para el cual se deberá dar clic en el botón Next de la aplicación, de la imagen 4.7. Lo cual realizara internamente dicho proceso, dando como resultado una ontología poblada con los metadatos de los OA del repositorio local.

#### **Módulo: Componente desarrollado y ontología poblada**

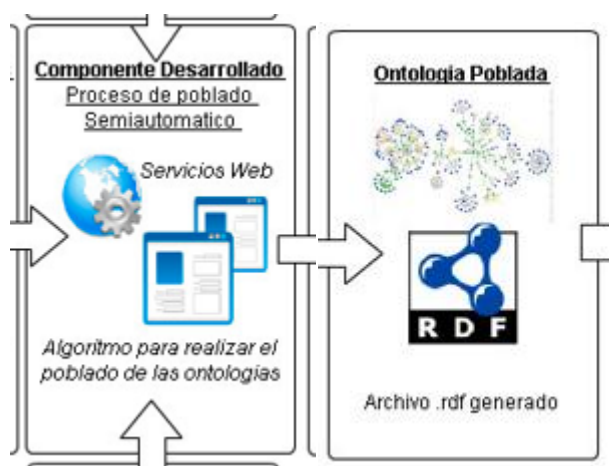


Imagen. 4.13 Módulos Componente desarrollado y ontología poblada

El componente desarrollado consiste en la aplicación del algoritmo para el proceso de poblado y para la generación de triples. Este componente reúne a los módulos descritos anteriormente, ya que el mismo utiliza los metadatos extraídos, las ontologías cargadas, así como también las reglas de mapeo que se proporcionaron a la aplicación, dando como resultado un archivo RDF que es la ontología poblada.

## Módulo: Vinculación de OA con recursos de DBPedia



Imagen. 4.14 Módulo: Vinculación de OA con recursos de DBPedia

Este proceso de vinculación es realizado a través del consumo de servicios web, lo cual se encuentra descrito en el capítulo 3, en la sección 3.8.2. Mediante DBpedia-Spotlight se enlaza los resultados obtenidos de consultas SPARQL (implementadas internamente en la aplicación), las cuales se procesan y devolverán como resultado los recursos relacionados, esto con el fin de enriquecer a los OA. Y para el presente estudio de caso el proceso de vinculación con fuentes de DBpedia ha devuelto como resultado enlaces generados por cada metadato de tipo subject de cada OA. A modo de ejemplo esto se puede ver en la sección de resultados del presente capítulo, en las tablas 4.4, 4.5, 4.6.

## Modulo: Almacenamiento de la ontología en Apache Marmotta

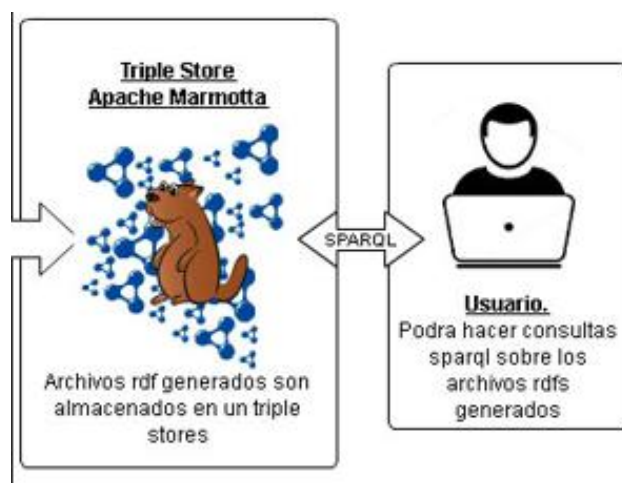


Imagen. 4.15 Módulo Almacenamiento en Apache Marmotta y consultas SPARQL



Esta es la parte final del proceso de poblado semiautomáticos de ontologías mediante la aplicación Metadata Harvester, y consiste en el almacenamiento de la ontología resultante, en un servidor Apache Marmotta, en el cual es posible realizar las consultas que se requieran, como se puede ver en la imagen 4.16, Marmotta permite realizar consultas a través de su terminal para este fin.

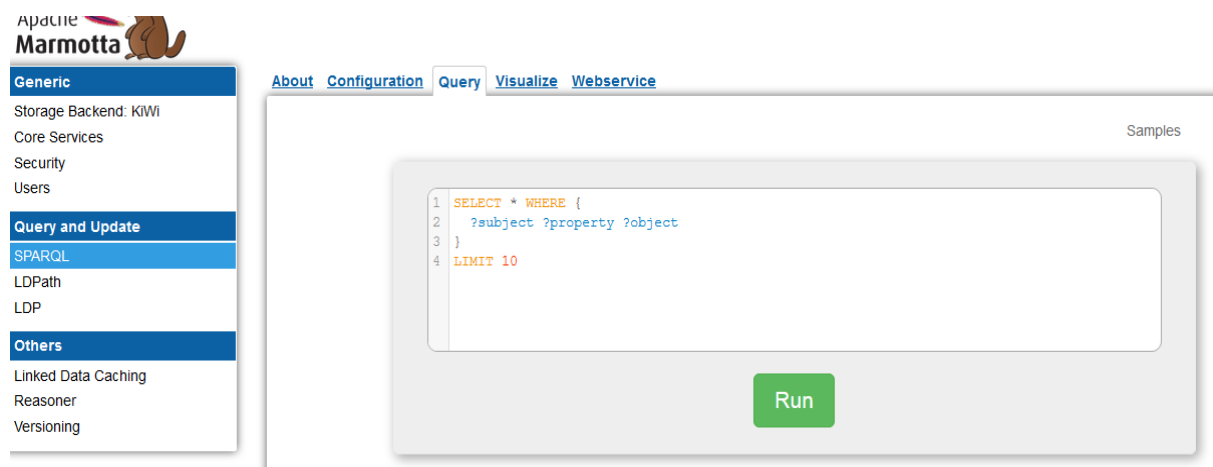


Imagen. 4.16 Consulta SPARQL sobre la ontología poblada

## Resultados

Una vez terminado el proceso de poblado se obtiene el tiempo que ha tomado el mismo, esto para todos los OA con los cuales se estaba trabajando. El cual se puede ver en la siguiente tabla.

Número de OA poblados	Número de pasos	Tiempo Total (Milisegundos)	Tiempo Total (segundos)
46	6	25830	00:00:25:830

Tabla 4.1 Resultados del proceso de poblado semiautomático

En resumen, el número de pasos que ha tomado este proceso es 6 y el tiempo total de poblado de los 46 OA ha sido de: 00:00:25:830. Y esto da como resultado una ontología poblada con los metadatos de los OA extraídos del repositorio local del pc1, lo cual se puede ver en la imagen 4.17, mediante la herramienta Protege

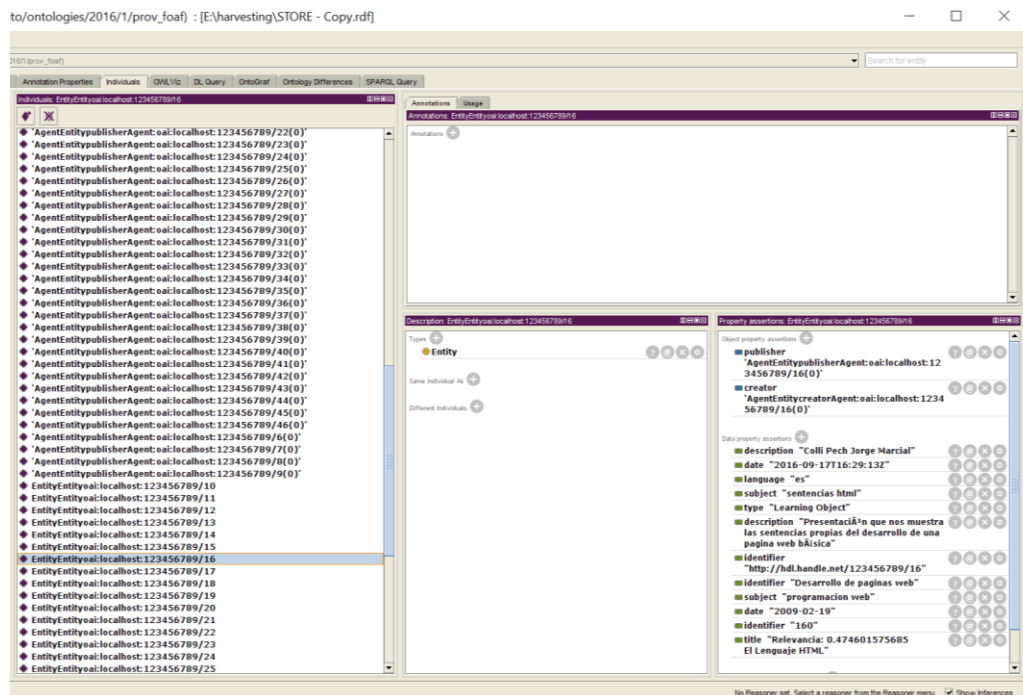


Imagen. 4.17 Visualización en Protege, de la ontología poblada con los metadatos

Además, la solución también ha realizado el proceso de vinculación con recursos de la fuente externa DBpedia, esto en base a los subjects que tiene cada OA poblado, lo cual se puede ver en las tablas 4.4, 4.5 y 4.6

## 4.3 Análisis de resultados

### 4.3.1 Respuestas a las preguntas del estudio de caso

Mediante el experimento controlado se ha podido dar respuesta a las preguntas planteadas para el estudio de caso

#### 1. ¿Cuál es el proceso de poblado de una ontología?

Poblar una ontología es el proceso de extraer y representar las clases, entidades y las relaciones entre los individuos y los valores de las propiedades. Este proceso está descrito con más detalle en el capítulo 2.

#### 2. ¿Cuántas maneras existen de poblar una ontología? ¿y cuáles son?

Existen 3 maneras de poblar una ontología: manual, semiautomática y automática.

##### a. ¿Cuántos OA se pueden poblar simultáneamente?

En el proceso manual se puede poblar solo un OA a la vez, mientras que en los procesos semiautomáticos y automáticos se pueden poblar 1 o varios OA

simultáneamente, y el tiempo de este proceso dependerá del número de OA con el cual se esté trabajando.

### **3. ¿Cuántos estándares existen para metadatos de OA?**

Existen varios estándares que permiten trabajar con metadatos de OA, de los cuales se analizaron en el capítulo 2 los siguientes: LOM, Dublin Core, SCORM, IMS.

#### **a. ¿Es posible poblar ontologías usando cualquier estándar?**

Existen estándares y ontologías específicas para el manejo de OA, como lo son LOM y LOM2OWL respectivamente, con los cuales es posible realizar el proceso de poblado sin mayor complicación, pero cuando se trabaja con otro tipo de estándares u ontologías que no son específicas para OA, es posible que se necesite realizar una mezcla de las mismas o algún proceso similar que permita poblar la ontología. Otro punto importante es contar con la ontología adecuada para el proceso de poblado. Por lo tanto, se puede responder a esta pregunta diciendo que, si es posible poblar ontologías usando cualquier estándar, pero su complejidad dependerá del estándar que se escoja.

### **4. Al agilizarse el proceso de poblado de ontologías ¿Cuánto tiempo se reduciría el mismo?**

La hipótesis que inicialmente se planteo es que la solución desarrollada reduciría notablemente el tiempo de poblado.

#### **a. ¿Cuánto tiempo se tarda en realizar el poblado de ontologías de forma manual?**

En el presente capítulo se ha procedido a realizar el poblado de una ontología, de forma manual. Mostrando a manera de ejemplo, el proceso que se realiza para la instanciación de dos individuos y una propiedad. En esta sección se presenta el número de pasos a seguir para realizar este proceso, en base a lo cual se presenta un tiempo estimado. Dado que es difícil determinar el tiempo exacto de un poblado manual, y considerando que un OA tiene muchas propiedades, de tipo DataProperties y ObjectProperties se ha intentado evitar cálculos subjetivos proponiendo un estimado de este tiempo, el cual es de 5 minutos por OA. Por lo tanto, esto implica que para

los 46 OA del repositorio local tomará un tiempo aproximado de 46 x 00:05:00 = 03:50:00.

**b. ¿Cuánto tiempo se tarda en realizar el poblado de ontologías usando la solución desarrollada?**

Mediante la aplicación que implemente la solución, el tiempo de poblado de la ontología, usando los metadatos de 46 OA extraídos de un repositorio local, es de 00:00:25:830, lo cual se mostró en la tabla 4.2. Además, los tiempos de poblado por OA se pueden ver en la siguiente imagen:

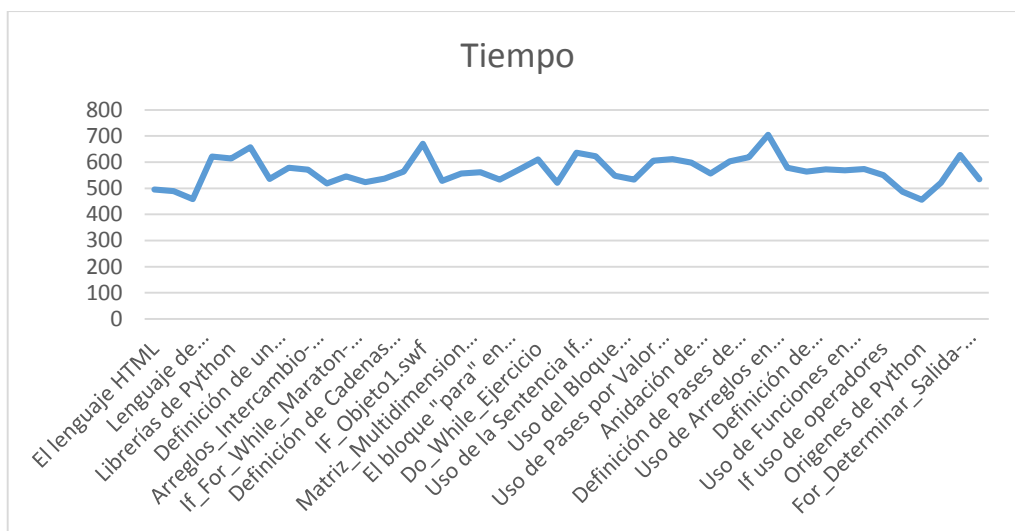


Imagen. 4.18 Tiempos de poblado por OA (tiempo en milisegundos)

**i. ¿Cuántas fuentes externas son vinculadas con los resultados de las consultas a las ontologías?**

El número de fuentes externas vinculadas depende de la cantidad de subjects que tiene cada OA, es por esta razón que no se puede dar un número exacto de fuentes externas vinculadas a los OA. Además de que existen OA con subjects que no se vincularon con ninguna fuente externa.

**ii. ¿Cuántas tripletas genero el proceso de poblado de la ontología?**

El número de tripletas que genero este proceso es de:

2758(1733 triples inicial)  
1025 triples creados

### c. ¿Cuándo se obtienen los mejores resultados?

Por los resultados obtenidos, se puede observar que en el proceso de poblado mediante la aplicación Metadata Harvester, se ha reducido considerablemente el número de pasos y por ende el tiempo de dicho proceso, por lo cual se puede concluir que se obtienen mejores resultados al usar la aplicación desarrollada.

### 4.3.2 Comparación entre los procesos de poblado: manual y semiautomático

La tabla 4.3 muestra una comparativa entre el proceso de poblado manual y el proceso semiautomático para la población de ontologías.

	Poblado Manual	Poblado mediante la solución
Número de OA poblados	1	46
Tiempo de poblado	Tiempo estimado (puede variar) 5 minutos por OA	00:00:25:830
Número de pasos	10 pasos por OA  Total: 46x10  460 pasos	6
Fuentes externas vinculadas	0	0 o más  Depende del número de subjects de cada OA
Número de ontologías que soporta	1	1 o mas
Número de individuos creados en la ontología	1	46 individuos principales
Número de DataProperties	1	10
Número de ObjectProperties	1	4
e	1751	2758

Vinculación con fuentes externas	No	Si (DBPedia)
----------------------------------	----	-----------------

En la tabla anterior se puede observar la diferencia que existe entre estos procesos, especialmente en el número de pasos y los tiempos de cada uno.

### 4.3.3 Vinculación con fuentes externas

Por otro lado, el proceso de vinculación con fuentes de DBpedia ha devuelto como resultado enlaces generados por cada metadato de tipo subject de cada OA. Por ejemplo, para un OA que tiene 3 subjects, se ha generado uno o más enlaces para cada uno de ellos. Además, es importante mencionar que ha existido algunos casos en los que no se ha generado enlaces con recursos de DBpedia. A continuación, se presentan los resultados para algunos OA:

**OA:** El Lenguaje HTML

**Subject:** web

<b>Título:</b>	El lenguaje HTML
<b>URL Dspace:</b>	<a href="http://www.w3.org/ns/prov#EntityEntityoai:localhost:123456789/7">http://www.w3.org/ns/prov#EntityEntityoai:localhost:123456789/7</a>
<b>subject:</b>	web
<b>URL Anotacion:</b>	<a href="http://spotlight.sztaki.hu:2222/rest/annotate?text=web&amp;confidence=0.2&amp;support=20">http://spotlight.sztaki.hu:2222/rest/annotate?text=web&amp;confidence=0.2&amp;support=20</a>
<b>URI:</b>	<a href="http://dbpedia.org/resource/World_Wide_Web">http://dbpedia.org/resource/World_Wide_Web</a>
<b>Stadistics:</b>	<a href="http://spotlight.sztaki.hu:2222/rest/candidates?text=web&amp;confidence=0.2&amp;support=20">http://spotlight.sztaki.hu:2222/rest/candidates?text=web&amp;confidence=0.2&amp;support=20</a>

Tabla 4.3 Enlaces DBpedia generados para el OA: El Lenguaje HTML que contiene un solo Subject

**OA** “Relación entre Apuntadores y Cadenas”

**Subject:** Relationship between pointer and strings

<b>Título:</b>	Relación entre Apuntadores y Cadenas
<b>URL Dspace:</b>	<a href="http://www.w3.org/ns/prov#EntityEntityoai:localhost:123456789/37">http://www.w3.org/ns/prov#EntityEntityoai:localhost:123456789/37</a>
<b>subject:</b>	Relationship between pointers and strings
<b>URL Anotacion:</b>	<a href="http://spotlight.sztaki.hu:2222/rest/annotate?text=Relationship+betwee">http://spotlight.sztaki.hu:2222/rest/annotate?text=Relationship+betwee</a>
<b>URI:</b>	<a href="http://dbpedia.org/resource/Relational_model">http://dbpedia.org/resource/Relational_model</a>
	<a href="http://dbpedia.org/resource/Pointer_(computer_programming)">http://dbpedia.org/resource/Pointer_(computer_programming)</a>
	<a href="http://dbpedia.org/resource/String_(computer_science)">http://dbpedia.org/resource/String_(computer_science)</a>
<b>Stadistics:</b>	<a href="http://spotlight.sztaki.hu:2222/rest/candidates?text=Relationship+betwee">http://spotlight.sztaki.hu:2222/rest/candidates?text=Relationship+betwee</a>

Tabla 4.4 Enlaces DBpedia generados para un OA titulado Relación entre Apuntadores y Cadenas, que contiene varios Subjects

**OA:** “Do\_While\_Ejercicio”

**Subject:** do while

No genera enlaces de DBpedia

<b>Título:</b>	<b>Do_While_Ejercicio</b>
<b>URL Dspace:</b>	Info: object: http://www.w3.org/ns/prov#EntityEntityoai:localhost:1234!
<b>subject:</b>	do while
<b>URL Anotacion:</b>	Info: http://spotlight.sztaki.hu:2222/rest/annotate?text=do+while&conf

Tabla 4.5 OA titulado Do\_While\_Ejercicio no genera enlaces de DBpedia

Una vez que se han generado los enlaces, se ha realizado un proceso de calificación manual, para determinar su validez, es decir, para determinar si estos ayudan o no al enriquecimiento del OA. Para esto, se han analizado los enlaces, su relación con el OA y con el tema global de los mismos (Programación), dando una calificación en base al criterio presentado en la tabla 4.7:

<b>Relación con el OA y con el tema (Programación)</b>	<b>Calificación manual</b>
No existe relación	0
Relación mediana	0.5
Relación fuerte	1

Tabla 4.6 Criterios de calificación manual

Todos los valores están dentro de estos criterios, es decir, han existido algunas calificaciones con valor 0.1, 0.4, 0.8, etc., dependiendo de la relación con el OA, pero todas caen dentro del rango de 0 a 1.

Se ha obtenido los promedios de relación, los cuales describen la correspondencia existente entre los enlaces de los Subjects y el tema de cada OA junto con el tema global: programación.

<b>Objetos de Aprendizaje</b>	<b>Relación con recursos de DBPedia</b>
Fundamentos de Programación	50%
Librerías de Python	15%



If/else diagrama de flujo	80%
Definición de un arreglo en ANSI C	70%
Orígenes de Python (SMIL 3)	30%
Arreglos_Intercambio-Ejercicio.swf	100%
El lenguaje HTML	73.33%
e	75%
For seleccionar opcion	50%
Definición de Cadenas en ANSI C	64%

Tabla 4.7 Relación entre los OA y los enlaces generados.

Este resultado de forma gráfica se puede ver en la imagen 4.19:

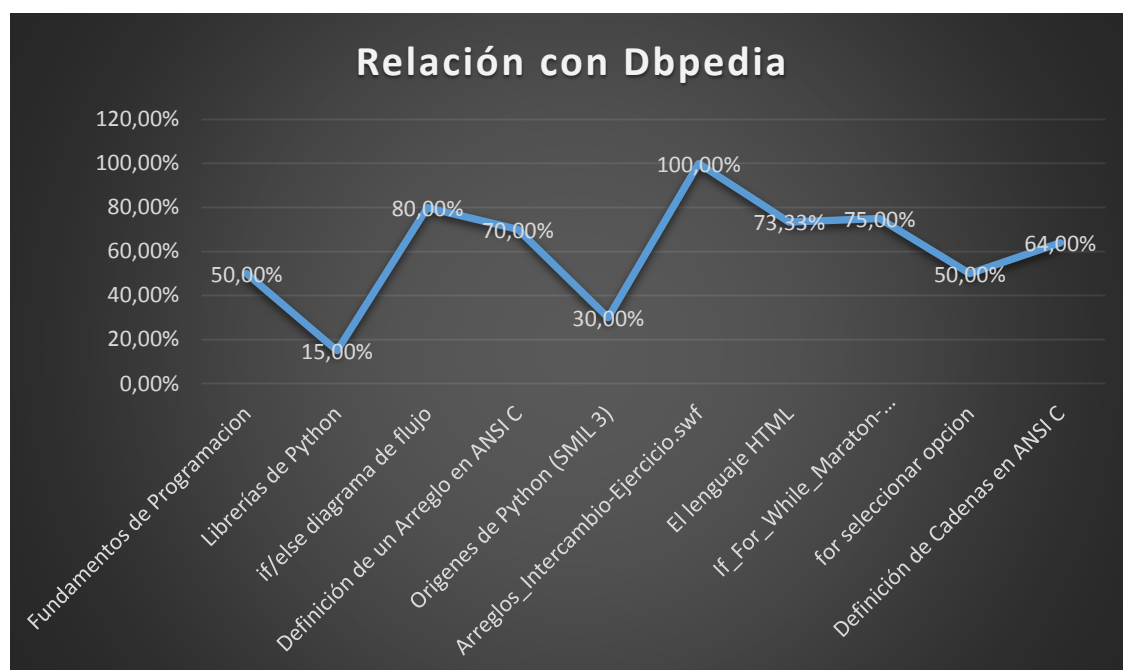


Imagen. 4.19 Gráfica de la relación entre OA y enlaces DBpedia

En la siguiente tabla se incluyen el número de Subjects y el número de enlaces para cada OA:

Objetos de Aprendizaje	Relación con recursos de DBPedia	Subjects	Links relacionados
Fundamentos de Programación	50%	2	2
Librerías de Python	15%	3	2
If/else diagrama de flujo	80%	6	3
Definición de un arreglo en ANSI C	70%	4	1
Orígenes de Python (SMIL 3)	30%	2	1
Arreglos_Intercambio-Ejercicio.swf	100%	3	3
El lenguaje HTML	73.33%	6	6
If_For_While_Maraton-Ejercicio.swf	75%	3	4
For seleccionar opcion	50%	3	3
Definición de Cadenas en ANSI C	64%	4	5

Tabla 4.8 Relación entre OA y enlaces DBpedia, Subjects y links

La siguiente gráfica muestra la relación entre el número de Subjects y el número de enlaces generados para cada OA:

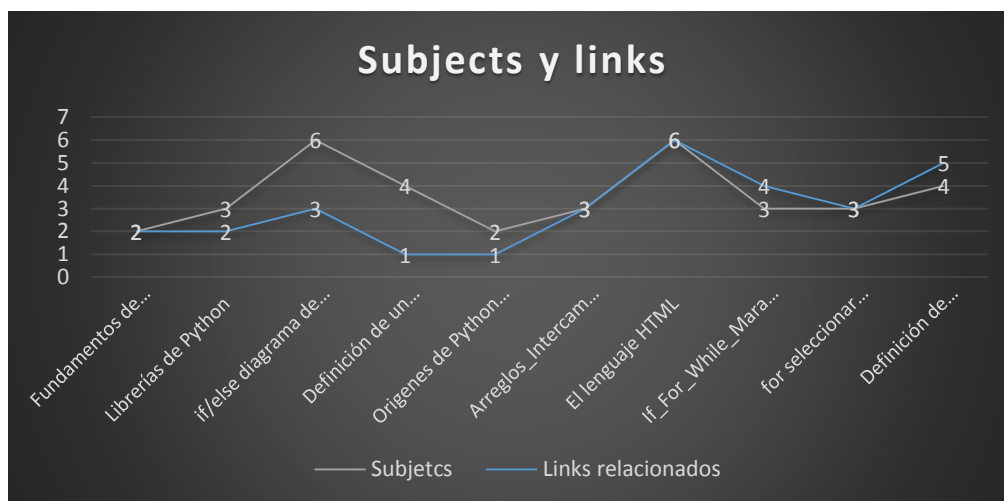


Imagen. 4.20 Gráfica que describe la relación entre Subjects y links de OA

El proceso de anotación mediante la solución desarrollada también ha devuelto estadísticas importantes de relación entre los OA y los enlaces de DBpedia, donde la estadística que engloba a todas las demás se denomina FinalScore, la cual

básicamente indica la relación entre el OA y los recursos DBpedia generados para el mismo. La siguiente tabla presenta los promedios de relación con FinalScore:

Objetos de Aprendizaje	Promedio de FinalScore(DBPedia)
Fundamentos de Programación	68.19%
Librerías de Python	98.18%
If/else diagrama de flujo	92.08%
Definición de un arreglo en ANSI C	91.90%
Orígenes de Python (SMIL 3)	99.06%
Arreglos_Intercambio-Ejercicio.swf	94.74%
El lenguaje HTML	94.92%
If_For_While_Maraton-Ejercicio.swf	90.15%
For seleccionar opcion	75.38%
Definición de Cadenas en ANSI C	81.14%

Tabla 4.9 Relación entre los OA y los enlaces, con promedios devueltos por FinalScore

Este resultado se puede ver de forma gráfica en la imagen 4.21. Y finalmente se tiene una comparativa entre los promedios de FinalScore y los promedios obtenidos mediante la calificación manual

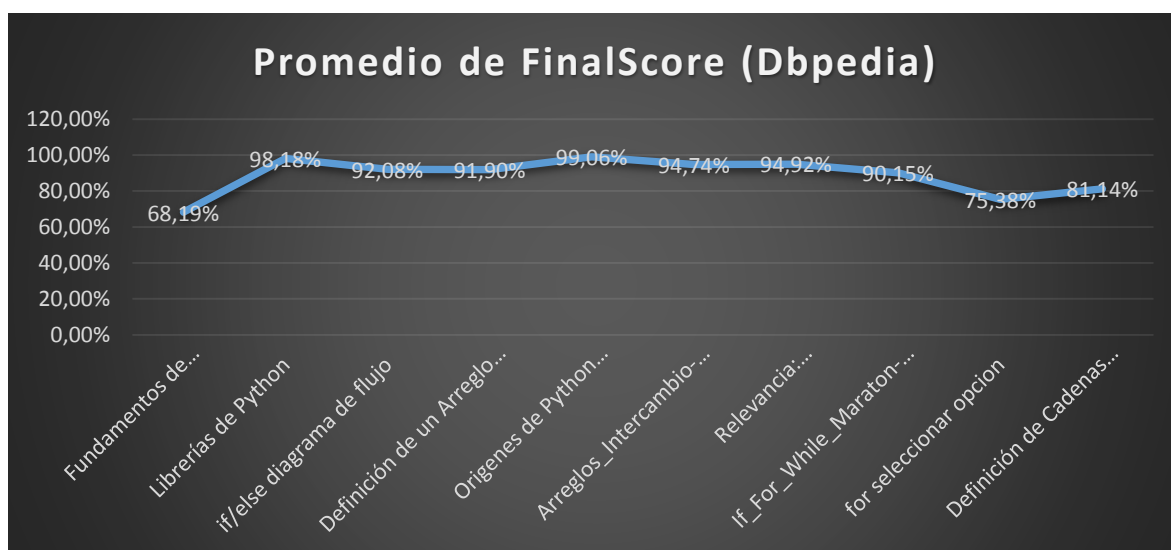


Imagen. 4.21 Gráfica de la relación entre los OA y los enlaces, con promedios devueltos por FinalScore.

Objetos de Aprendizaje	Promedio de FinalScore(DBPedia)	Promedio de Calificación manual
Fundamentos de Programación	68.19%	50%
Librerías de Python	98.18%	15%
If/else diagrama de flujo	92.08%	80%
Definición de un arreglo en ANSI C	91.90%	70%
Orígenes de Python (SMIL 3)	99.06%	30%
Arreglos_Intercambio-Ejercicio.swf	94.74%	100%
El lenguaje HTML	94.92%	73.33%
If_For_While_Maraton-Ejercicio.swf	90.15%	75%
For seleccionar opcion	75.38%	50%
Definición de Cadenas en ANSI C	81.14%	64%

Tabla 4.10 Promedios de FinalScore y promedios de calificación manual

Este resultado de forma gráfica se puede ver en la siguiente imagen:

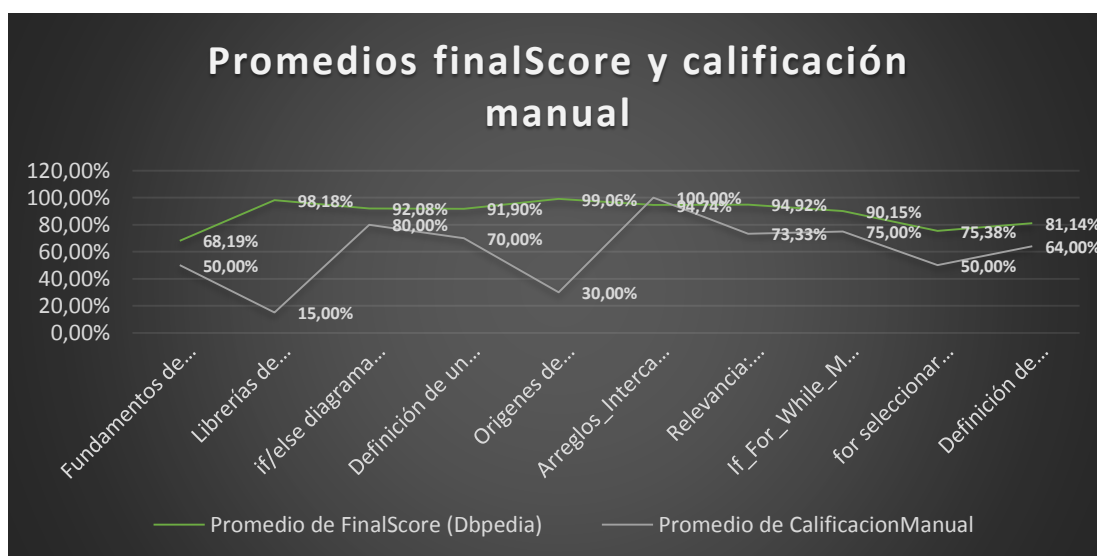


Imagen. 4.22 Promedios comparativos entre calificación manual y estadística de DBpedia

## 4.4 Interpretación de resultados y conclusiones del experimento

Como se puede apreciar en la tabla 4.3 existe una clara diferencia, principalmente en cuanto al tiempo de poblado de las ontologías. Esta tabla presenta la comparación de los resultados entre el poblado manual de un OA frente al poblado semiautomático implementado en la aplicación Metadata Harvester. Además, es importante notar que el tiempo de poblado manual para un OA que se muestra en la tabla 4.3 es únicamente un estimado por experiencia propia, mas no es un tiempo exacto, ya que el mismo dependerá de varios factores mencionados en las secciones anteriores. Este tiempo estimado es de 5 minutos por OA, esto implica que ese tiempo crecerá conforme al número de OA que se pretendan poblar en la ontología, es decir, para los 46 OA del repositorio local implicaría un tiempo estimado de:

$$00:05:00 \times 46 = 03:50:00$$

Esto hace que la aplicación Metadata Harvester tenga una validez muy superior, ya que mediante la misma el tiempo de poblado se reduce a:

$$00:00:25:830$$

Las tablas 4.4, 4.5 y 4.6 muestran los enlaces de DBpedia generados para los Subjects de los OA. Se presenta a modo de ejemplo estas 3 tablas, pero para todos los OA se tiene un resultado similar. Se puede apreciar que el número de enlaces dependerá del número de Subjects de cada OA. Además, se han presentado casos en los que no se han generado enlaces, como lo muestra la tabla 4.6

La tabla 4.7 presenta los criterios de calificación para la relación entre cada enlace generado y los OA. Estos criterios se han establecido de forma manual, y por ende esta calificación también ha sido un proceso manual. Obteniéndose finalmente los promedios presentados en la tabla 4.8 y la imagen 4.19, que describe estas relaciones. Mientras más alta sea la calificación, mejor será la relación encontrada entre el OA y los recursos de DBpedia.

La tabla 4.9 y la imagen 4.20 presentan el número de subjects y el número de enlaces por OA. Esto es importante, ya que permite conocer cuántos subjects tiene un OA y cuantos enlaces se encuentran para cada uno de ellos. Y es probable que mientras más enlaces existan, mejor sea la calificación de la relación entre el OA y

los recursos de DBpedia. Aunque también existe la posibilidad de que algunos enlaces no enriquezcan al contenido del OA.

La tabla 4.10 y la imagen 4.21 presentan los promedios de relación basados en la estadística denominada FinalScore, la misma que es devuelta por DBpedia, y básicamente mide la relación entre los OA y los recursos, es decir, consiste en el proceso descrito en el punto anterior, pero ya no mediante una calificación manual.

Finalmente se tiene una comparación entre los promedios manual y finalScore, que se presenta la tabla 4.11 y la imagen 4.22. Es importante indicar que el tiempo del proceso de poblado semiautomático, mediante la aplicación Metadata Harvester, dependerá de los equipos que se usen y de la velocidad de la red con la cual se esté trabajando. Por lo tanto, este tiempo puede variar.

## 4.5 Recapitulación

En este capítulo se realizó la ejecución de un experimento controlado como caso de estudio, en donde se ha evidenciado las ventajas que ofrece la aplicación Metadata Harvester, especialmente en el tiempo de poblado de las ontologías. Dentro de este experimento se manejó un escenario controlado, que consistía en un repositorio local con 46 OA, al cual se accedió mediante el protocolo OAI-PMH y se extrajo sus metadatos, con los cuales se ha realizado el poblado de una ontología determinada, mediante el algoritmo implementado en la aplicación Metadata Harvester. Como medio para poder realizar el análisis y la comparación e interpretación de los de los resultados se utilizó el proceso descrito en el capítulo 2, en el cual se describen los procesos de poblado de una ontología.

Como resultado de este experimento controlado, se puede concluir que la aplicación Metadata Harvester ofrece beneficios especialmente en el tiempo de poblado de ontologías. Además de brindar la posibilidad de realizar consultas SPARQL sobre dichas ontologías, esto en el escenario de un triple store como Apache Marmotta. Por otro lado, la aplicación vincula los resultados de las consultas SPARQL implementadas, con recursos de DBpedia.

# Capítulo 5. Conclusiones y trabajos futuros

## 1.1 Conclusiones

Las conclusiones obtenidas del presente proyecto se exponen a continuación:

En el capítulo 3 del presente proyecto se ha definido tres tipos de poblado de ontologías: manual, semiautomático y automático. El proceso manual es demasiado largo y ocupa un tiempo excesivo. El proceso semiautomático es mucho más ágil, pero requiere la presencia de un experto (persona), para establecer las reglas de mapeo, mientras que el proceso automático no requiere de dicho experto, lo que lo convierte en un proceso más eficiente, aunque su implementación es más compleja. Sin embargo, aunque la aplicación Metadata Harvester muestra considerables ventajas frente a un proceso manual, este último fue esencial para comprender el proceso de poblado de un OA en una ontología.

Los resultados obtenidos en el experimento controlado, presentado en el capítulo 4 han demostrado la validez de la aplicación Metadata Harvester, como producto final del presente proyecto. Se han utilizado 46 OA para realizar el proceso de poblado de los mismos, y se ha evidenciado el número de pasos del poblado manual (10 pasos por OA, total: 460 pasos) y del poblado semiautomático (6 pasos), así como también los tiempos de poblado manual (aproximadamente 03:50:00) y semiautomático (exactamente: 00:00:25:830). Demostrando la validez de la solución desarrollada, debido a su eficiencia en la reducción del número de pasos y por ende en los tiempos de poblado.

La finalidad de generar enlaces para los OA, es que los mismos enriquezcan su contenido, lográndose con esto tener un contenido más alto y de mejor calidad. Los enlaces generados para cada OA dependerán del número de Subjects y del número de palabras de cada Subject. Mientras más enlaces se generen, más alta será la probabilidad de enriquecer al OA. Existen casos en los que para un determinado OA no se generan enlaces a recursos DBpedia, como se evidencio en el capítulo 4, en la sección de resultados.



El proceso de calificación de la relación entre un OA y sus recursos DBpedia generados, permite determinar la validez de los mismos, es decir, permite determinar si estos enlaces enriquecen o no al OA. Este proceso se lo ha desarrollado de forma manual, y se ha concluido que existen OA que tienen una relación alta con los recursos de DBpedia, como es el caso del OA titulado: Arreglos\_Intercambio-Ejercicio.swf el cual presenta una calificación del 100%, mientras que existen otros como el OA titulado Librerías de Python que tiene una calificación de 15%. Para este proceso de calificación se ha considerado únicamente la revisión manual de cada enlace, sin embargo, pueden existir otros factores que alteren estos resultados.

DBpedia devuelve estadísticas de relación entre sus recursos y los Subjects de un OA. Los promedios de estas estadísticas están representados en una variable denominada FinalScore, la cual determina este nivel de relación. Los valores para esta estadística están entre 0 y 1, siendo 1 el nivel más alto y 0 el más bajo. Los promedios de calificación manual y los devueltos por DBpedia que se obtuvieron muestran grandes variaciones en ciertos OA, como es el caso de OA titulado Librerías de Python y Orígenes de Python (SMIL 3). Sin embargo, en los demás OA, la gráfica de la imagen 422 presenta un comportamiento similar.

## 1.2 Líneas de trabajo futuras

Dentro del marco de desarrollo del presente proyecto se encuentran las siguientes líneas de trabajo futuras:

- Implementación del proceso automático de poblado de ontologías, el mismo que no requiere la presencia de un experto.
- Realizar experimentos controlados para medir el tiempo de población de ontologías mediante un proceso automático.
- Analizar y definir métricas que permitan realizar una comparación del rendimiento y eficiencia de los distintos tipos de poblado: manual, semiautomático y automático.
- Realizar vinculación de los resultados con otras fuentes externas a más de DBpedia.
- Adaptar la solución desarrollada para que trabaje con más protocolos de comunicación, lo que permitirá abarcar un mayor número de repositorios y recursos de la Web.
- Mejorar la precisión en cuanto a las anotaciones que permitan la vinculación con fuentes externas, como DBpedia.

# Glosario

## Mapping

Proceso de mapeo para el poblado de una de una ontología, mediante el cual se establecen reglas para el manejo de los metadatos en ontologías.

## IMS

IMS es un consorcio que agrupa a vendedores, productores, implementadores y consumidores de e-learning. Este se enfoca completamente a desarrollar especificaciones en formato XML. (Hernández, 2001)

## Endpoint

Punto de entrada a un servicio, el cual puede ser por ejemplo un servicio de consultas SPARQL.

## Linked Data

Son datos enlazados o datos vinculados que describen un método de publicación de datos estructurados para que puedan ser interconectados y más útiles.

## URL

Son las siglas de Uniform Resource Locator (Localizador Uniforme de Recursos) y se trata de la secuencia de caracteres que permite denominar recursos dentro de la Internet, para que estos puedan ser localizados.

## URI

Son las siglas de Uniform Resource Identifier (Identificador Uniforme de Recursos). Y consiste en una cadena de caracteres que se utiliza para identificar un nombre o un recurso dentro de la Internet.

## IRI

Son las siglas de Identificadores de Recursos Internacionales. Y básicamente este es un nuevo estándar de internet que extiende a las URI. Mientras que las

URIs usan subconjuntos de carácter ASCII, las IRIs pueden contener caracteres Unicode como caracteres chinos o japoneses

## URN

URN son las siglas de Uniform Resource Name, que sirve también para identificar a un recurso en una red a través de un nombre.

# Anexos

## Instalación y configuración de Dspace

### Prerrequisitos de software

Dspace necesita de los siguientes requisitos de software para ser ejecutado en un equipo con sistema operativo Windows 7.

- OpenJDK 6 o 7  
(<http://www.oracle.com/technetwork/java/javase/downloads/index.html>).
- PostgreSQL en versiones comprendidas entre 8.3 a 8.4  
(<https://www.postgresql.org/download/>).
- Apache Tomcat 5.5 o superior  
(<http://tomcat.apache.org/download-60.cgi>).
- Apache Maven 2.2 o superior  
(<http://maven.apache.org/download.cgi>).
- Apache Ant 1.8 o superior  
(<http://ant.apache.org/bindownload.cgi>).
- Dspace 3.1  
(<https://sourceforge.net/projects/Dspace/files/>)

### Instalación de software necesario

Para el proceso de instalación se debe seguir el siguiente procedimiento:

1. El proceso inicia con la descarga e instalación del JDK descargado del enlace indicado en el apartado anterior. La ruta por defecto de instalación de java es C:\Program Files. Posterior a la instalación es necesario configurar el PATH 'C:\Program Files\Java\jdk1.6.0\_39\bin' y la variable de entorno

JAVA\_HOME 'C:\Program Files\Java\jdk1.6.0\_39\bin'. La ruta especificada varía de acuerdo a la versión de JDK elegida.

2. Una vez instalado java se procede a la instalación de Apache Ant y Apache Maven, ambas son distribuciones binarias por lo que se las debe descomprimir en un directorio del equipo. Posteriormente se debe agregar el PATH correspondiente para Ant 'C:\apache-ant-1.8.4\bin' y Maven 'C:\apache-maven-2.2.1\bin'. Finalmente se configura la variable de entorno ANT\_HOME 'C:\apache-ant-1.8.4\bin'. La ruta especificada en la configuración de Apache Ant y Apache Maven dependerán de las versiones elegidas para instalar.
3. El siguiente paso es la descarga e instalación de la base de datos PostgreSQL. Una vez instalado la base de datos es necesario crear el rol Dspace y la base de datos del mismo nombre.
4. Posteriormente se debe llevar a cabo la instalación del servidor Apache Tomcat mediante el instalador de Windows (archivo exe). Una vez instalado este se incluirá dentro de la lista de servicios de Windows desde donde puede ser iniciado o detenido, por defecto Apache Tomcat se desplegará en la dirección <http://localhost:8080>, pero esta dirección puede ser configurable.
5. Finalmente se realizará la instalación de Dspace, para lo cual se debe descargar el código fuente del enlace incluido en el apartado anterior y descomprimirlo en un directorio del equipo (Por ejemplo, C: \). Posterior creamos el directorio para la instalación (Por ejemplo, C: Dspace\)

## Instalación de Dspace

Una vez descargado e instalado todo el software es necesario realizar la siguiente configuración para desplegar Dspace.

1. Dentro del directorio de instalación de Dspace en el archivo de configuración 'Dspace-3.1-src-release\Dspace\config\Dspace.cfg' se debe configurar las siguientes propiedades como se muestra en la ilustración 1:
  - **Dspace.dir**: directorio de instalación de Dspace.
  - **Dspace.url**: URL complete de la página de Dspace en el servidor.
  - **Dspace.hostname**: nombre del dominio del servidor web.
  - **Dspace.name**: nombre asignado al servidor.
  - **db.password**: contraseña de la base de datos asignada en la instalación de postgresQL.

```
##### Basic information #####

# DSpace installation directory
dspace.dir = C:/dspace

# DSpace host name - should match base URL. Do not include port number.
dspace.hostname = 172.27.4.145

# DSpace base host URL. Include port number etc.
dspace.baseUrl = http://172.27.4.145:8080

# DSpace base URL. Include port number etc., but NOT trailing slash
# Change to xmlui if you wish to use the xmlui as the default, or remove
# "/jspui" and set webapp of your choice as the "ROOT" webapp in
# the servlet engine.
dspace.url = ${dspace.baseUrl}/xmlui

# Optional: DSpace URL for mobile access
# This
dspace.mobileUrl = http://mobile.example.com

# Name of the site
dspace.name = ${dspace.name}

# Default language for metadata values
default.language = ${default.language}

##### Database settings #####

# Database name ("oracle", or "postgres")
db.name = postgres

# URL for connecting to database
db.url = jdbc:postgresql://localhost:5432/dspace

# JDBC Driver
db.driver = org.postgresql.Driver

# Database username and password
db.username = dspace
db.password = dspace
```

Ilustración 1 Configuración de propiedades en el archivo Dspace.cfg. Fuente: Autores.

2. Se procede a generar el paquete de instalación de Dspace mediante la ejecución del comando 'mvn package' dentro del directorio de del código fuente descargado como se muestra en la ilustración 2:

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\swebshet>cd C:\dspace-3.1-src-release\dspace
C:\dspace-3.1-src-release\dspace>mvn package
```

Ilustración 2 Comando de generación del paquete de instalación. Fuente: Autores.

Una vez ejecutado el comando se generará varios paquetes que se utilizan en los pasos siguientes.

3. En el directorio 'C:\Dspace-3.1-src-release\dspace\target\Dspace-3.1-build' generado en el paso anterior se procede a ejecutar el comando 'ant fresh install' como se muestra en la imagen ilustración 3. Es importante recordar que para ejecutar este comando es necesario haber iniciado la base de datos.

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\swebshet>cd \
C:\>cd C:\dspace-3.1-src-release\dspace\target\dspace-3.1-build
C:\dspace-3.1-src-release\dspace\target\dspace-3.1-build>ant fresh install
```

Ilustración 3 Comando de ejecución de instalación. Fuente: Autores.

4. En el directorio C:\Dspace\bin\ se procede a ejecutar el comando 'Dspace create-administrator' como se muestra en la imagen ilustración 4, esto para crear la cuenta administrador. Se ingresa la información solicitada durante la instalación; correo electrónico, nombre, apellido y contraseña.

```
C:\dspace\bin>dspace create-administrator
Using DSpace installation in: C:\dspace
Creating an initial administrator account
E-mail address:
First name:
Last name:
WARNING: Password will appear on-screen.
Password:
Again to confirm:
Is the above data correct? (y or n): y
Administrator account created
```

Ilustración 4 Creación de cuenta administrador. Fuente: Autores.

5. Finalmente se procede a copiar las carpetas jspui, xmlui y oai del directorio donde se instaló Dspace 'C:/Dspace/webapps' dentro de la carpeta donde se instaló anteriormente el servidor web 'C:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps'. Una vez realizado este paso e iniciado el servidor web se podrá acceder a Dspace a través de las URLs <http://localhost:8080/jspui/> y <http://localhost:8080/xmlui/>

## Creación del repositorio local de OA en Dspace

Una vez que ha finalizado la instalación de Dspace, se tiene la comunidad principal que es Dspace y dentro de la misma se procede a la creación de una colección, la cual será el repositorio local que contendrá a los ítems de la misma, que en este caso son los OA para el presente experimento.

Para la creación de una colección se procede a través del botón que muestra la imagen siguiente, la misma que conducirá a la ilustración 5, en donde se deberá llenar los campos correspondientes a la nueva colección

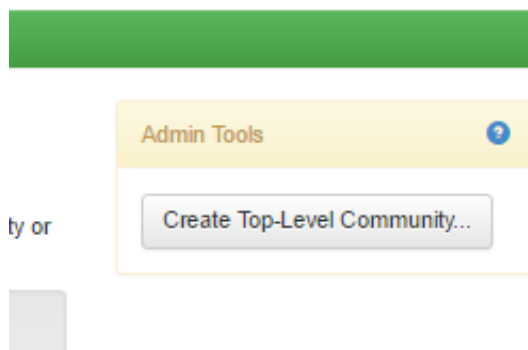


Ilustración 5 Botón para crear una nueva colección den Dspace

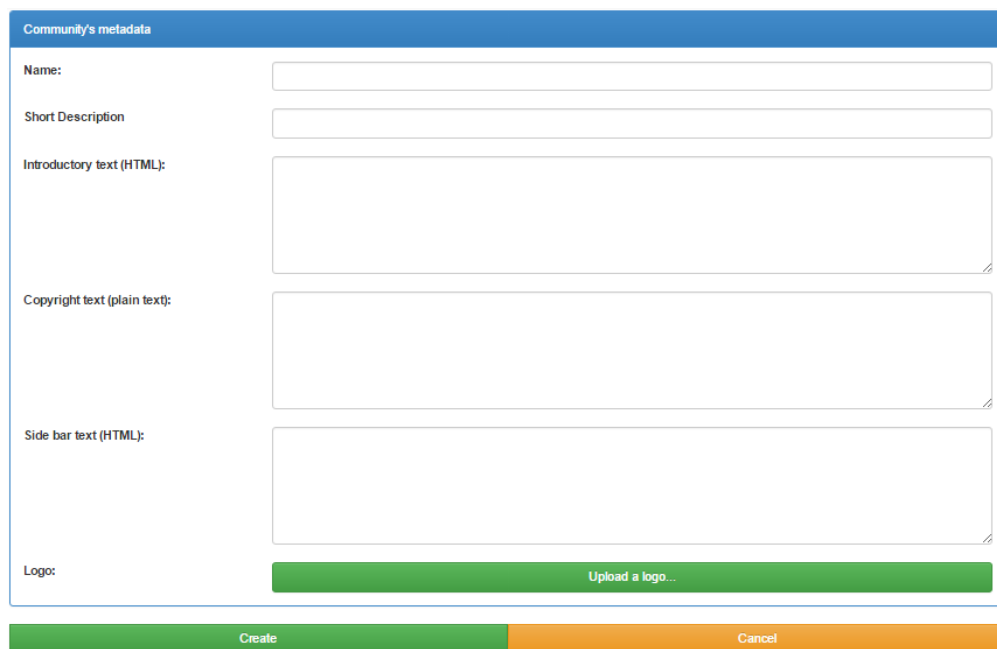


Ilustración 6 Campos correspondientes a una nueva colección

La ilustración 7 presenta la interfaz que presenta Dspace como comunidad y dentro de la misma la colección para los OA.

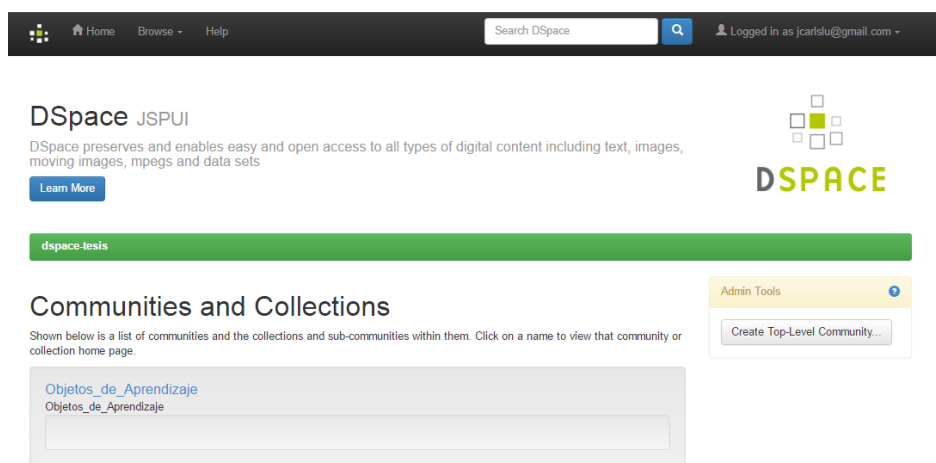


Ilustración 7 Interfaz de Dspace. Repositorio local para Objetos de Aprendizaje (Dspace)

## Subir los OA al repositorio de OA

Ahora se procede a subir los OA al repositorio creado. Para lo cual se sigue el siguiente proceso:

### Paso 1.



Dentro de la colección se da clic en la opción de agregar recurso (OA), lo cual abrirá la pantalla de la ilustración 8. Donde se procede a llenar los campos que se muestran, los mismos que permiten describir al OA.

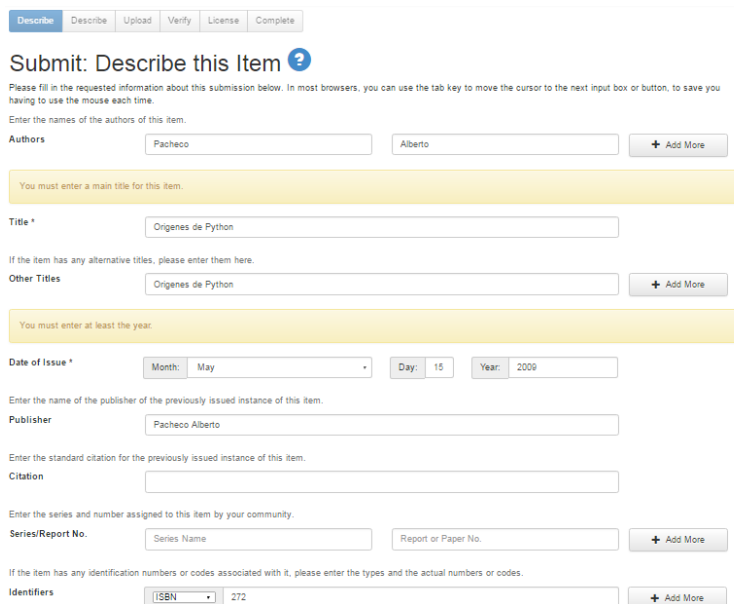


Ilustración 8 Publicación de un OA en el repositorio de Dspace (1)

## Paso 2.

Se procede a llenar los campos necesarios como autor, titulo, keywords, descripción, entre otros, los cuales se convierten en metadatos que describen al OA que se está subiendo al repositorio.

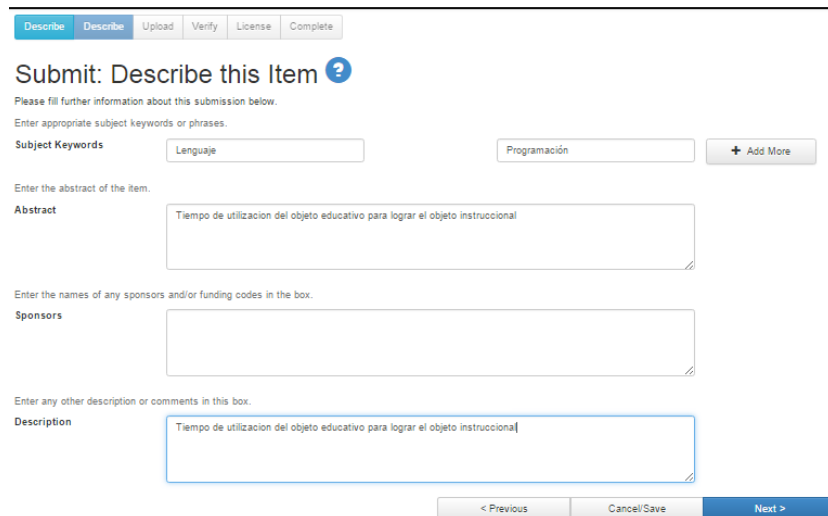
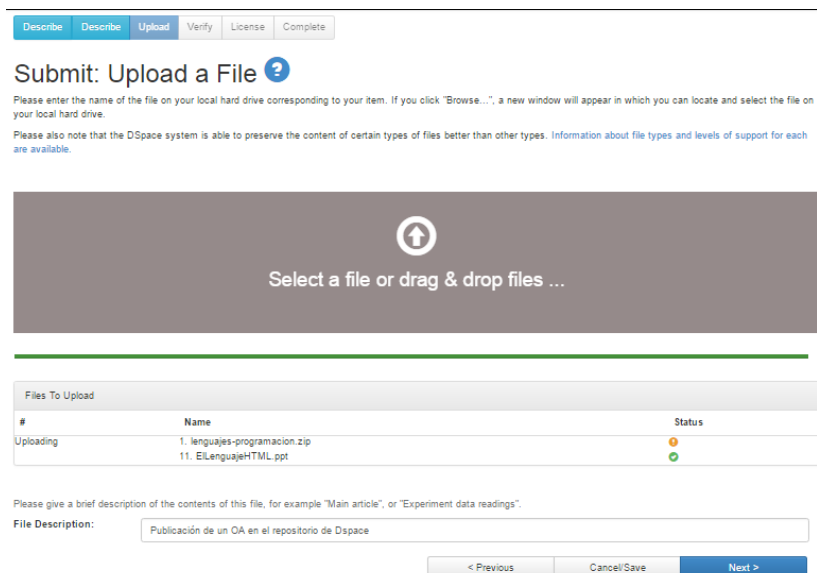


Ilustración 9 Publicación de un OA en el repositorio de Dspace (2)

### Paso 3.

La ilustración 10 muestra la pantalla donde se debe seleccionar el archivo que contiene al OA que se va a subir al repositorio de OA



Describe Describe Upload Verify License Complete

### Submit: Upload a File ?

Please enter the name of the file on your local hard drive corresponding to your item. If you click "Browse...", a new window will appear in which you can locate and select the file on your local hard drive.

Please also note that the DSpace system is able to preserve the content of certain types of files better than other types. [Information about file types and levels of support for each are available.](#)

Select a file or drag & drop files ...

#	Name	Status
Uploading	1. lenguajes-programacion.zip	⚠
	11. EILenguajeHTML.ppt	✅

Please give a brief description of the contents of this file, for example "Main article", or "Experiment data readings".

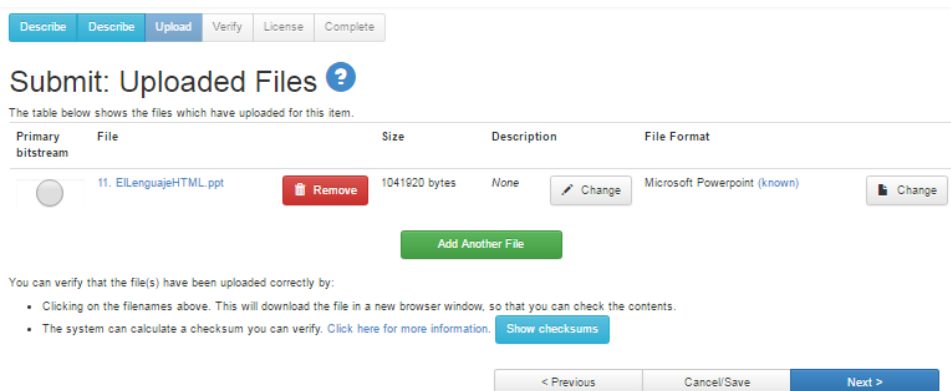
File Description:

< Previous Cancel/Save Next >

Ilustración 10 Selección del archivo correspondiente al OA en el repositorio de Dspace (3)

### Paso 4.


En la pantalla que muestra la ilustración 11 se verifica que el archivo se haya subido de forma correcta.



Describe Describe Upload Verify License Complete

### Submit: Uploaded Files ?

The table below shows the files which have uploaded for this item.

Primary bitstream	File	Size	Description	File Format
	11. EILenguajeHTML.ppt <a href="#">Remove</a>	1041920 bytes	None <a href="#">Change</a>	Microsoft Powerpoint (known) <a href="#">Change</a>

[Add Another File](#)

You can verify that the file(s) have been uploaded correctly by:

- Clicking on the filenames above. This will download the file in a new browser window, so that you can check the contents.
- The system can calculate a checksum you can verify. [Click here for more information.](#) [Show checksums](#)

< Previous Cancel/Save Next >

Ilustración 11 Publicación de un OA en el repositorio de Dspace (4)

### Paso 5.

Finalmente se verifica la información correspondiente al OA. Esto se muestra en la ilustración 12 y 13, donde se puede ver una pantalla de confirmación para subir el OA al repositorio

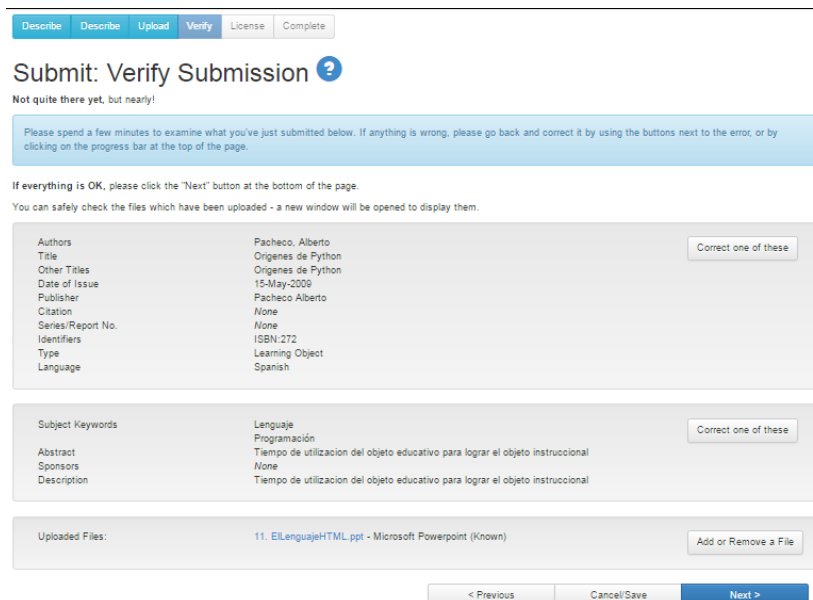


Ilustración 12 Publicación de un OA en el repositorio de Dspace (5)

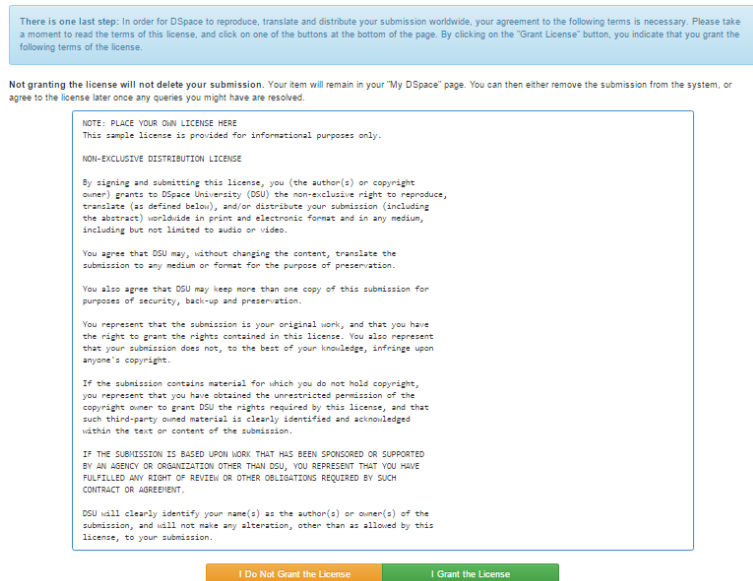


Ilustración 13 Publicación de un OA en el repositorio de Dspace (6)

## Paso 6.

La ilustración 14 muestra que se subió con éxito el OA en el repositorio de Dspace. Con lo cual se ha publicado exitosamente este OA en el repositorio Dspace local.

Describe
Describe
Upload
Verify
License
Complete

### Submit: Submission Complete!

Your submission will now go through the workflow process designated for the collection to which you are submitting. You will receive e-mail notification as soon as your submission has become a part of the collection, or if for some reason there is a problem with your submission. You can also check on the status of your submission by going to the My DSpace page.

[Go to My DSpace](#)  
[Communities and Collections](#)

Submit another item to the same collection

Ilustración 14 Publicación de un OA en el repositorio de Dspace (7)

Para el presente experimento se realiza la publicación de 46 OA, con el fin de utilizar este número de OA para realizar el poblado de una ontología mediante la solución desarrollada. La ilustración 15 muestra los OA en el repositorio de Dspace

Issue Date	Title	Author(s)
3-Aug-2009	Uso de la Sentencia For en ANSI C	Castro Magaña, Jesús Aurelio
3-Aug-2009	Uso de Funciones en ANSI C[Producto y División]	Castro Magaña, Jesús Aurelio
3-Aug-2009	Uso de Funciones en ANSI C[Factorial y Combinación]	Castro Magaña, Jesús Aurelio
3-Aug-2009	Definición de Apuntadores en ANSI C	Castro Magaña, Jesús Aurelio
3-Aug-2009	Uso de Cadenas en ANSI C	Castro Magaña, Jesús Aurelio
3-Aug-2009	Uso de Arreglos en ANSI C	Castro Magaña, Jesús Aurelio
3-Aug-2009	Sintaxis y semántica en ANSI C	May Chan, José Luis
3-Aug-2009	Definición de Pases de Datos en ANSI C	Castro Magaña, Jesús Aurelio
3-Aug-2009	Relación entre Apuntadores y Cadenas	Castro Magaña, Jesús Aurelio
4-Aug-2009	Anidación de Sentencias de Control de Flujo	Castro Magaña, Jesús Aurelio
3-Aug-2009	Uso de Pases por Valor y por Referencia[2]	Castro Magaña, Jesús Aurelio
3-Aug-2009	Uso de Pases por Valor y por Referencia[1]	Castro Magaña, Jesús Aurelio
3-Aug-2009	Uso del Bloque "Mientras que"	Castro Magaña, Jesús Aurelio
3-Aug-2009	El bloque "Mientras que" en diagramas de flujo	Castro Magaña, Jesús Aurelio
3-Aug-2009	Uso del Bloque "Decisión"	Castro Magaña, Jesús Aurelio
3-Aug-2009	Bloque "Decisión" en ANSI C	Yichao, Zhu
3-Aug-2009	Uso de la Sentencia If en ANSI C	Castro Magaña, Jesús Aurelio
4-Aug-2009	Bloque "mientras que" en ANSI C	Castro Magaña, Jesús Aurelio, Castro Magaña, Jesús Aurelio
4-Aug-2009	Do_While_Ejercicio	Gómez, Sergio
3-Aug-2009	continue.swf	Castellanos Cáceres, Tomás Antonio Manuel

Gómez, Sergio 3  
Jorge Marcial, Colli Pech 3  
Pacheco, Alberto 3  
Cambranes, Edgar 1  
Castro Magaña, Jesús Aurelio, Cast... 1  
Colli Pech, Jorge Marcial 1  
May Chan, José Luis 1  
Yichao, Zhu 1  
next >

Subject
ANSI C 27  
Programación estructurada 20  
programación estructurada 6  
sentencias de control 6  
Programación 4  
Arreglos 3  
Definición de los Pases por valor... 3  
Desarrollo 3  
Diagramas de flujo y 3

Ilustración 15 Listado de OA publicados en el repositorio de Dspace.

## Proceso manual de poblado de una ontología.

Probablemente este es el proceso más importante a tener en cuenta, debido a que el mismo permite entender el concepto de creación de individuos, instancias, propiedades y relaciones entre individuos.

A continuación, se ha procedido a poblar una ontología de forma manual. Los metadatos se han tomado del primer OA que da como resultado la búsqueda en el repositorio de Ágora, como se muestra en la ilustración 16, mediante el criterio de búsqueda de “programación”. La descripción del OA seleccionado para este proceso se presenta en la tabla 1:

**Listar Recursos**

Para buscar un recurso escribe una palabra o frase que lo identifica, también puedes usar el [Visor de metadatos](#) para definir una palabra de búsqueda.

Texto:  [Búsqueda avanzada...](#)

☐ En la red de servidores AGORA (puede ser tardado)

Recursos listados: 46

Tipo	Recurso	Kb	Publicador	Opciones	Herramientas
	<a href="#">Orígenes de Python</a> Podcast multimedia MP3 Introducción a un Lenguaje de Programación usando la metáfora de anuncio de película de cine. Versión para iPod (iTunes) y PC (RealPlayer 11 o QuickTime). Estadísticas: 0 <a href="#">lenguajes-programacion.zip</a>	2,977.95	Pacheco Alberto	   	
	<a href="#">El lenguaje HTML</a> Programacion en HTML Archivo de recurso para programacion en HTML Estadísticas: 1 <a href="#">HTML2.ppt</a>	80.00	Colli Pech Jorge Marcial	   	
	<a href="#">Lenguaje de programacion para paginas web</a> Desarrollo Web Archivo de recursos html Estadísticas: 1 <a href="#">html.doc</a>	78.50	Colli Pech Jorge Marcial	   	
	<a href="#">Lenguaje de programacion</a> Desarrollo Web Archivo de recurso para el desarrollo de páginas web usando código html	96.84	Colli Pech Jorge	   	

Ilustración 16 Resultado de búsqueda en ROA Ágora por criterio: programación

Información del OA	
OA:	Orígenes de Python
Identificador:	272



Título:	Orígenes de Python
Comentario	Podcast multimedia MP3
Descripción	Introducción a un Lenguaje de Programación usando la metáfora de anuncio de película de cine. Versión para iPod (iTunes) y PC (RealPlayer 11 o QuickTime).
Usuario:	Pacheco Alberto (162)
Archivo:	lenguajes-programacion.zip
Ubicación:	Local
Extensión:	Zip
Texto:	lenguajes-programacion.zip.txt
Propiedades:	lenguajes-programacion.zip.dat
LOM:	lenguajes-programacion.zip.lom
LO:	lenguajes-programacion.zip.zip
Tamaño:	3049416
<b>1. General</b>	
Identificador. Catálogo:	AGORA
Identificador. Entrada:	272
Título:	Orígenes de Python
Idioma:	Es
Palabra clave:	Lenguaje, Programación,
Estructura:	Linear
Nivel de agregación:	1
<b>2. Ciclo de vida</b>	



Versión:	v 1.0
Estado:	Final
Contribución. Tipo:	technical validator
Contribución. Entidad:	Pacheco Alberto
Contribución. Fecha:	15/06/2009
<b>3. Meta-metadatos</b>	
Contribución. Tipo:	Creator
Contribución. Entidad:	Alberto Pacheco alberto@acm.org
Contribución. Fecha:	15/06/2009 10:43
<b>4. Técnico</b>	
Formato:	audio/mpeg
Tamaño:	3 Mb
Localización:	<a href="http://161.67.140.11/agora/recurso/ver/contenido/272">http://161.67.140.11/agora/recurso/ver/contenido/272</a>
Pautas de instalación:	specific software Installation
Duración:	2 a 3 min
<b>5. Uso educativo</b>	
Tipo de interactividad:	Expository
Idioma:	Es
Tipo de recurso educativo:	Slide
Nivel de interactividad:	Low



Densidad semántica:	Médium
Destinatario:	Learner
Contexto:	higher education
Dificultad:	Médium
Tiempo típico de aprendizaje:	10 minutos

Tabla 1 Descripción del OA a poblar en la ontología

## Proceso

### Paso 1.

Abrir Protege y cargar la ontología LOM2OWL.owl. Para lo cual se debe dar clic en el menú File, y elegir la opción Open, la cual abrirá el cuadro de dialogo que se muestra en la ilustración 17, donde se deberá escoger la ontología que se va a cargar y dar clic en el botón Abrir.

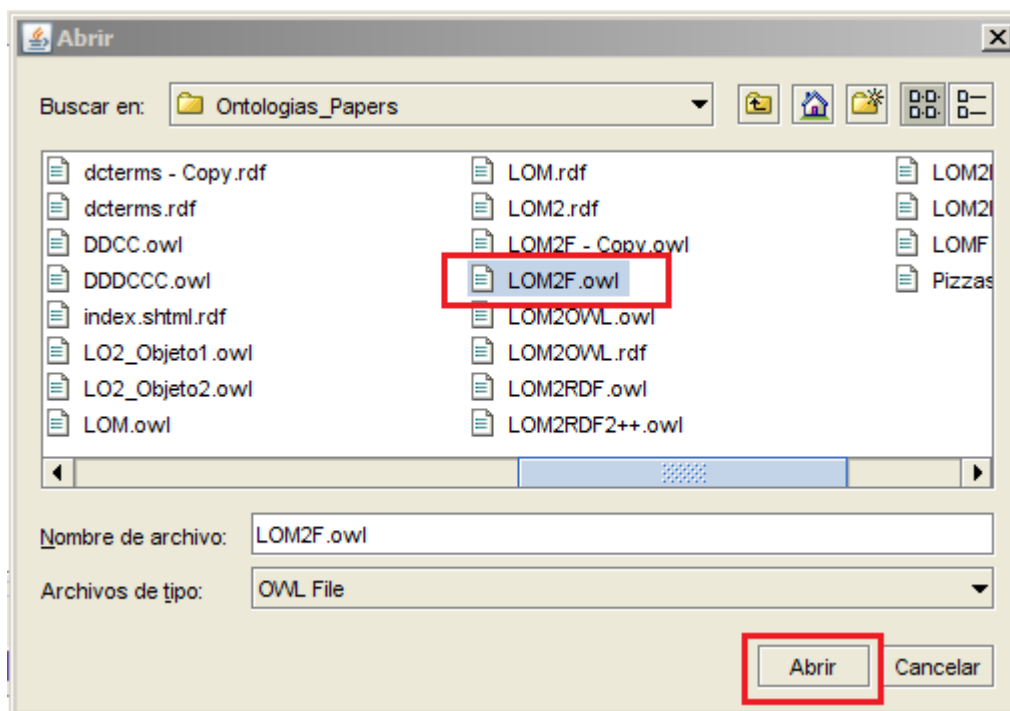


Ilustración 17 Cuadro de dialogo para seleccionar ontología para cargar en Protege

Esto cargará la ontología en Protege y permitirá ver todos los elementos relacionados a la misma, como las entidades, clases, propiedades, etc., esto en cada uno de las pestañas de la Protege, como lo muestra la ilustración 18:

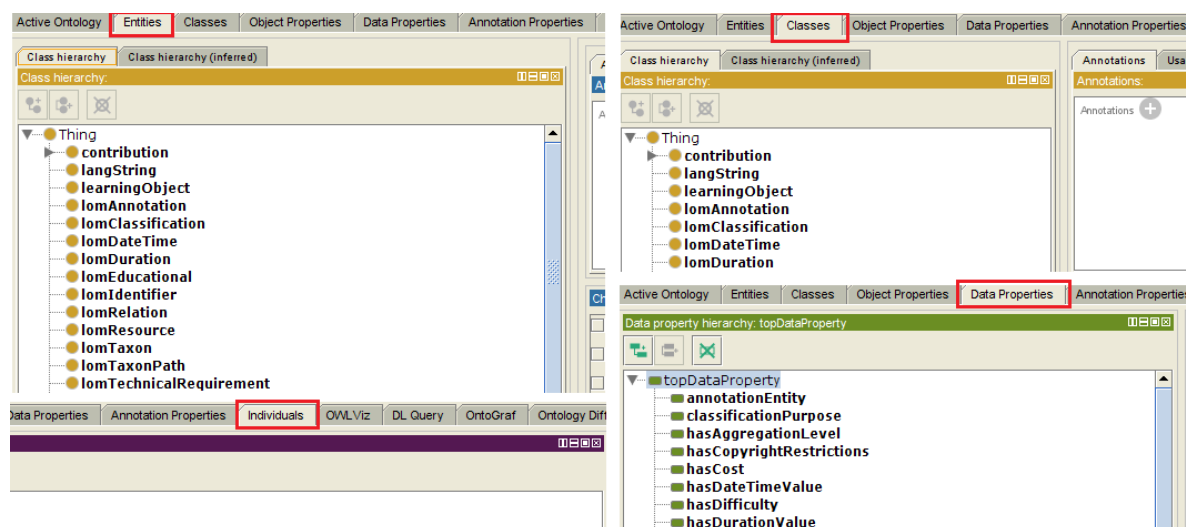


Ilustración 18 Clases, entidades, propiedades e individuos de la ontología cargada

## Paso 2.

Clic en la pestaña “Individuals” para la creación que permitirá mostrar los individuos creados:



Ilustración 19 Pestañas de trabajo de Protege

En primera instancia la ontología no tiene ningún individuo o instancia, como lo muestra la ilustración 20. Aquí es donde ha procedido a la creación de individuos.

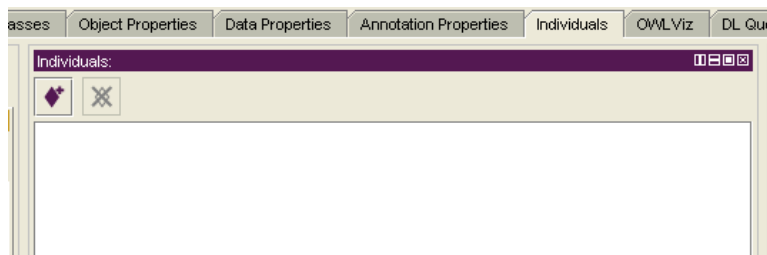



Ilustración 20 Ontología sin individuos

### Paso 3.

Para crear un nuevo individuo. dar clic en el icono , el cual abre el siguiente cuadro de dialogo:

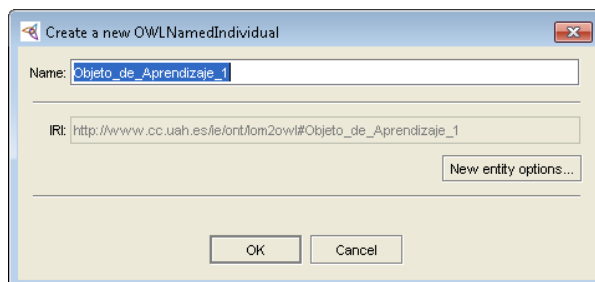


Ilustración 21 Cuadro de dialogo, crear nuevo individuo

En la pantalla anterior se debe llenar los campos para el nuevo individuo. Es importante indicar que el nombre del individuo preferiblemente no debe contener espacios en blanco, ya que si los tiene Protege los cambiara por “\_”. Así mismo, en esta pantalla se puede apreciar la IRI que tendrá este nuevo individuo, como se observa en la imagen siguiente:

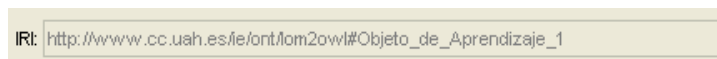


Ilustración 22 IRI del individuo “Objeto\_de\_Aprendizaje\_1”

### Paso 4.

Una vez que se haya creado la instancia del individuo, se ha procedido a agregarle el tipo al cual pertenece este individuo. Para ello se da clic en el icono de la descripción de individuos. El cual abre la pantalla que se muestra en la ilustración 23

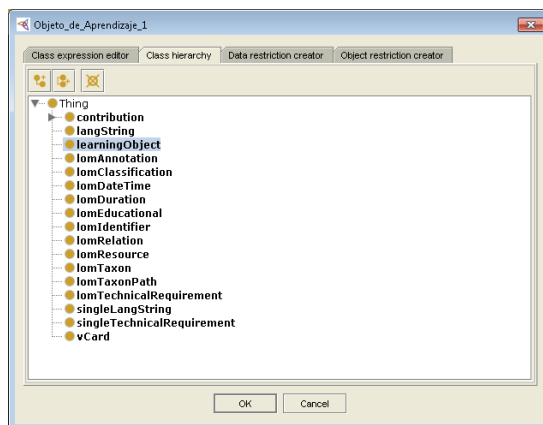


Ilustración 23 Cuadro de dialogo para escoger el tipo al que pertenece el individuo

## Paso 5.

Elección de la clase correspondiente.

Aquí se ha escogido el tipo de la clase a la cual pertenece el individuo, en este caso es de tipo `learningObject`.



Ilustración 24 Elección de la clase correspondiente para el individuo

## Paso 6.

El siguiente paso es agregar las propiedades del individuo: "Objeto\_de\_Aprendizaje\_1". Los cuales pueden ser Object Property o Data Property. Los primeros son propiedades que relacionan al primer individuo con otros individuos mediante propiedades (por ejemplo: `hasIdentifier`) mientras que los segundos son propiedades directas que se dan al individuo "Objeto\_de\_Aprendizaje\_1" (por ejemplo: `title`). Como se puede apreciar en las ilustraciones 25 y 26 el dominio y el rango de los properties es diferente según sea el caso. En el caso de los Object Properties estos serán clases de la ontología, y en el caso de los Data Properties, el dominio será una clase, mientras que el rango será un tipo de dato.

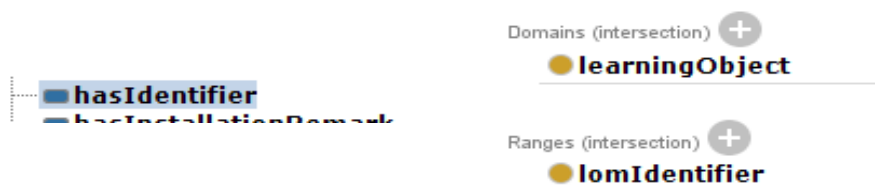


Ilustración 25 Dominio y Rango de un Object Property (`hasIdentifier`)



Ilustración 26 Dominio y Rango de un Data Property (`hasLanguage`)

Básicamente, el proceso de poblado de un OA consiste en instanciar un individuo (Objeto de Aprendizaje) con todos los elementos (propiedades) de las 9 categorías

descritas en el Anexo XII “EQUIVALENCIAS CASTELLANO-INGLÉS LOM-ES V1.0/LOM Y ANÁLISIS COMPARATIVO SOBRE CARÁCTER CATEGORÍAS Y ELEMENTOS DE DATOS” (Gobierno de España, 2009)

### Paso 7.


A continuación, se ha procedido a agregar el identificador al individuo “Objeto\_de\_Aprendizaje\_1”. Para lo cual se ha creado un nuevo individuo para identificador, esto debido a que la propiedad que relaciona al OA con su identificador es de tipo Object Property. Este proceso es el mismo que se ha seguido para el primer individuo creado.

Al individuo identificador se ha agregado el tipo y también los correspondientes DataProperties que describan su valor.

### Paso 8.

Agregar los dataProperties

Para agregar los DataProperties a este individuo se ha procedido de la siguiente manera:

Clic en el icono  el cual abrirá la pantalla mostrada en la ilustración 27, donde se deberá seleccionar el Data Property.

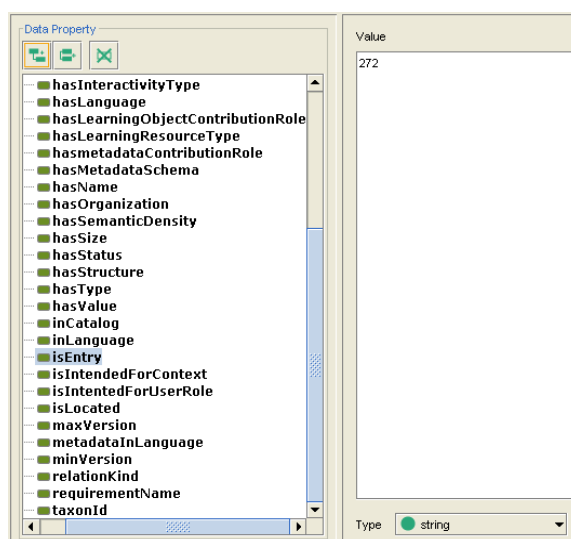


Ilustración 27 Selección del DataProperty (IsEntry y inCatalog) para el individuo Identificador

## Paso 9.

Darle un valor al data property seleccionado. En la pantalla que muestra la ilustración 28 se ha seleccionado el tipo de dato (string, doublé, float, etc.) y se escribe su valor. Esto se realiza cada vez que se agrega un DataProperty a un individuo y con esto se forma una tripleta.

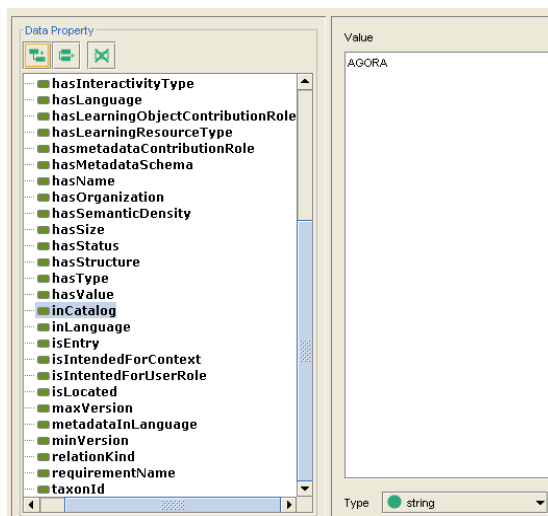


Ilustración 28 Valor para el data property seleccionado

Entonces el individuo para el identificador queda de la siguiente manera:

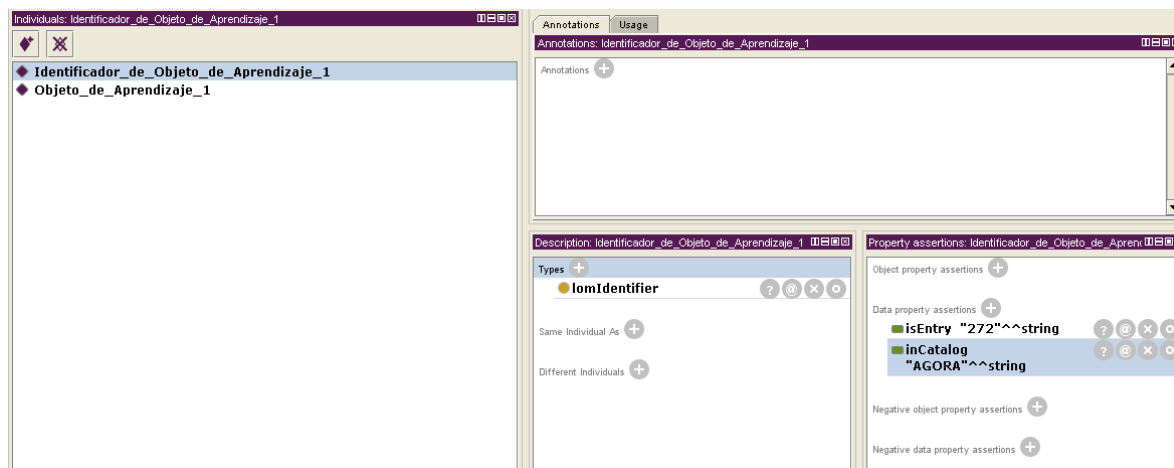


Ilustración 29 Individuo Identificador creado en Protege

## Paso 10.

Relacionar los individuos. Ahora se procede a relacionar el individuo "Identificador" con el individuo "Objeto\_de\_Aprendizaje\_1". Para esto se señala este último individuo y se da clic en el icono **Object property assertions** lo cual abre la pantalla que se

muestra en la ilustración 30, donde se escoge el individuo para establecer la relación mediante el un ObjectProperty. En otras palabras, aquí se forman las tripletas entre estos individuos.

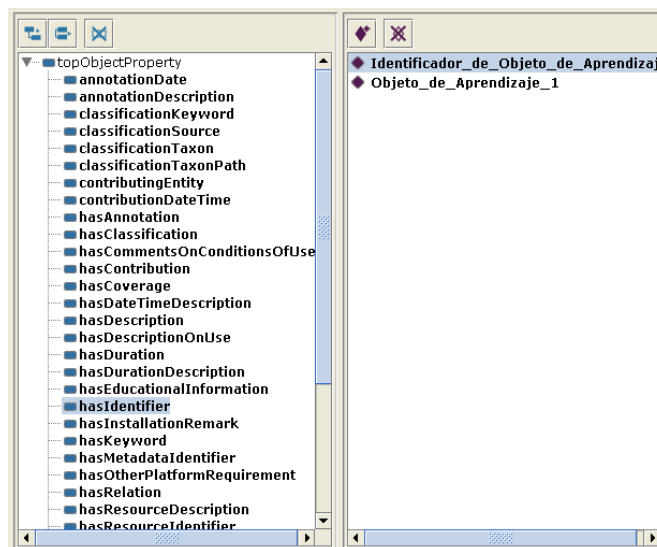


Ilustración 30 Relación entre los individuos mediante el Object Property

Finalmente, el individuo “Objeto\_de\_Aprendizaje\_1” queda de la siguiente forma:

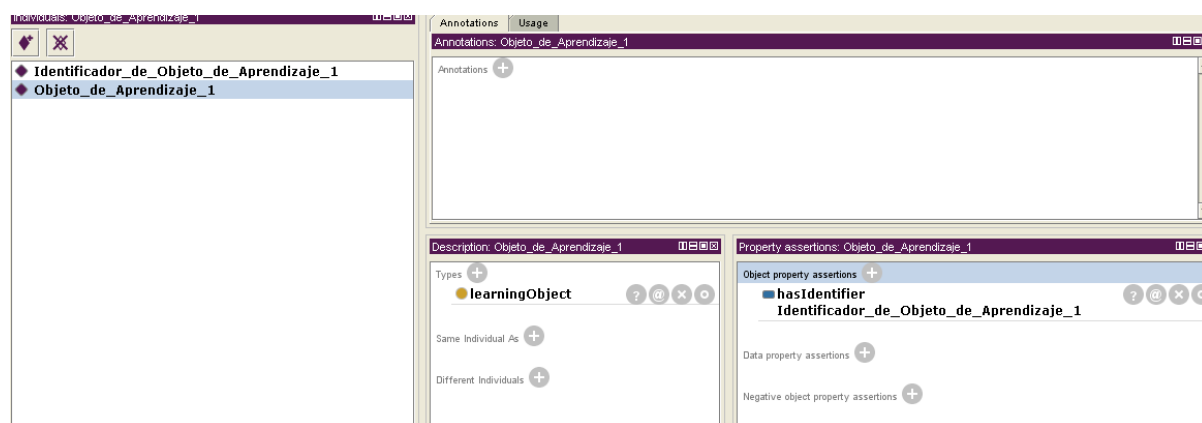


Ilustración 31 Individuo Objeto de Aprendizaje creado y relacionado con su identificador

El individuo principal, en este caso el “Objeto\_de\_Aprendizaje\_1”, tendrá todos los ObjectProperty y DataProperty necesarios. Por lo tanto, el proceso descrito para su creación será el mismo para todos los casos.



La siguiente imagen muestra un listado con todas las propiedades de un individuo en Protege.



Ilustración 32 Propiedades de un OA en Protege

Es importante señalar que el proceso descrito consigue poblar con un OA a la ontología LOM2OWL, esto quiere decir que el mismo proceso se realizara según el número de OA que se quieran poblar en la ontología. Por lo tanto, se realizará los 8 pasos anteriores por cada OA. El proceso descrito anteriormente dependerá de la ontología con la cual se esté trabajando, y también de la experiencia del usuario que esté realizando este proceso de poblado. Por lo tanto, es difícil determinar un tiempo exacto que tome este proceso, pero por experiencia propia para la ontología usada en el ejemplo descrito anteriormente se estima un tiempo aproximado de 5 minutos por OA.

## Referencias

- Agudelo Benjumea, M. M. (2008, Diciembre). Los metadatos. Recuperado a partir de [http://aprendeenlinea.udea.edu.co/lms/men/docsoac3/0301\\_metadatos.pdf](http://aprendeenlinea.udea.edu.co/lms/men/docsoac3/0301_metadatos.pdf)
- Apache Marmotta. (2013). Recuperado a partir de <http://marmotta.apache.org/>
- Area Moreira, M., & Adell Segura, J. (2009). e-Learning: Enseñar y Aprender en Espacios Virtuales. Recuperado a partir de <http://tecedu.webs.ull.es/textos/eLearning.pdf>
- Benítez, H., & Robayo, S. (2006). PROTOCOLO Z39.50 UNA HERRAMIENTA IMPORTANTE EN LA RECUPERACIÓN DE LA INFORMACIÓN.
- Berlanga, A. J., López, C., Morales, E., & García, F. J. (2011). Consideraciones para Reforzar el Valor de los Metadatos en los Objetos de Aprendizaje. Recuperado a partir de <http://www.uoc.edu/symposia/spdece05/pdf/ID03.pdf>
- Brickley, D., & Miller, L. (2014, enero 14). FOAF Vocabulary Specification 0.99 Namespace Document 14 January 2014 - Paddington Edition. Recuperado a partir de <http://xmlns.com/foaf/spec/>
- Cajal, F., Moraga, P., & Rosanigo, B. (2012). Un Repositorio de Objetos de Aprendizaje. Recuperado a partir de [http://sedici.unlp.edu.ar/bitstream/handle/10915/23685/Documento\\_completo.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/23685/Documento_completo.pdf?sequence=1)
- Canabal, M., Sarasa, A., & Sacristán, J. C. (2006). LOM-ES: Un perfil de aplicación de LOM. Recuperado a partir de [http://www.web.upsa.es/spdece08/contribuciones/110\\_Spedece2008-lom-es.pdf](http://www.web.upsa.es/spdece08/contribuciones/110_Spedece2008-lom-es.pdf)
- Daiber, J., Jakob, M., Hokamp, C., & Mendes, P. N. (2013). Improving Efficiency and Accuracy in Multilingual Entity Extraction. *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*. Recuperado a partir de

<https://pdfs.semanticscholar.org/196d/7e0a43587b0216c47a1ddb8cb1875aa1c1d0>  
.pdf

del Moral, E., & Cernea, D. A. (2005). Diseñando Objetos de Aprendizaje como facilitadores de la construcción del conocimiento. Recuperado a partir de <http://www.uoc.edu/symposia/spdece05/pdf/ID16.pdf>

Dublin Core to PROV Mapping. (2013). Recuperado a partir de <https://www.w3.org/TR/prov-dc/>

ECSDI. (2015). Lenguajes de Ontologías. Web Semántica. Recuperado a partir de <http://www.cs.upc.edu/~bejar/ecsd/Teoria/ECSDI05c-Lenguajes.pdf>

Fermoso García, A. M., Sánchez Alonso, S., & A. Sicilia, M. (2008). Una ontología en OWL para la representación semántica de objetos de aprendizaje. Recuperado a partir de [http://www.web.upsa.es/spdece08/contribuciones/176\\_Fermoso\\_Sanchez\\_Sicilia\\_LOMOWL.pdf](http://www.web.upsa.es/spdece08/contribuciones/176_Fermoso_Sanchez_Sicilia_LOMOWL.pdf)

Gobierno de España. (2009, junio 17). ANEXO XII EQUIVALENCIAS CASTELLANO-INGLÉS LOM-ES V1.0/LOM Y ANÁLISIS COMPARATIVO SOBRE CARÁCTER CATEGORÍAS Y ELEMENTOS DE DATOS. Recuperado a partir de <http://educalab.es/documents/10180/40863/LOM-ESejemplo12.pdf/c1607ba6-36c4-4439-b28e-64107d63f4d7>

Gómez Fierros, J. D. (2012, noviembre 30). *Poblado Automático de Ontologías Espaciales a Partir de Texto no Estructurado*. Centro Nacional de Investigación y Desarrollo Tecnológico Departamento de Ciencias Computacionales, Mexico. Recuperado a partir de <http://www.cenidet.edu.mx/subplan/biblio/seleccion/Tesis/MC%20Juan%20Diego%20Gomez%20Fierros%202012.pdf>



- Han, Q., & Gao, F. (2012, agosto 23). Towards semantic learning object metadata: mapping standard metadata specifications to ontologies. Recuperado a partir de <http://ieeexplore.ieee.org/document/6360326/?reload=true>
- Hernández, E. (2001). Estándares y Especificaciones de E- learning: Ordenando el Desorden. Recuperado a partir de <http://www.uv.es/ticape/docs/eduardo.pdf>
- Hilera González, J. R., & Hoya Marín, R. (2010). ESTÁNDARES DE E-LEARNING: GUÍA DE CONSULTA. Recuperado a partir de <http://www.cc.uah.es/hilera/GuiaEstandares.pdf>
- Jovanović, J., Gašević, D., & Torniai, C. (2016). Learning Object Context Ontologies - The LOCO Framework. Recuperado a partir de <http://jelenajovanovic.net/LOCO-Analyst/loco.html>
- Kiryakov, A., Popov, B., Ognyanoff, D., Manov, D., Kirilov, A., & Goranov, M. (2003). Semantic Annotation, Indexing, and Retrieval. Recuperado a partir de [http://ontotext.com/documents/publications/2003/SemAIR\\_ISWC169.pdf](http://ontotext.com/documents/publications/2003/SemAIR_ISWC169.pdf)
- Laufer, C. (2015). Guia de la Web Semántica. Recuperado a partir de <http://ceweb.br/guias/web-semantica//es/>
- Lendinez Teixeira, S. (2009). *APLICACIÓN PARA LA POBLACION DE ONTOLOGÍAS*. Carlos III de Madrid, Madrid. Recuperado a partir de [http://e-archivo.uc3m.es/bitstream/handle/10016/8522/PFC\\_Sebastian\\_Lendinez\\_Teixeira.pdf?sequence=1&isAllowed=y](http://e-archivo.uc3m.es/bitstream/handle/10016/8522/PFC_Sebastian_Lendinez_Teixeira.pdf?sequence=1&isAllowed=y)
- López, M. G., Miguel, V., & Montaña, N. (2000). Sistema Generador de AMBientes de EnseñanzaApRendizaje Constructivistas basados en Objetos de Aprendizaje (AMBAR), 10.
- López Guzmán, C., & García Peñalvo, F. J. (2009). Repositorios de objetos de aprendizaje: bibliotecas para compartir y reutilizar recursos en los entornos e-learning. Recuperado a partir de

[http://gredos.usal.es/jspui/bitstream/10366/21700/1/GRIAL\\_Repositoriosobjetosaprendizaje.pdf](http://gredos.usal.es/jspui/bitstream/10366/21700/1/GRIAL_Repositoriosobjetosaprendizaje.pdf)

López Guzmán, C., García Peñalvo, F., & Pernías Peco, P. (2005, Enero del). Desarrollo de repositorios de objetos de aprendizaje a través de la reutilización de los metadatos de una colección digital: de Dublin Core a IMS. Portal Universia. Recuperado a partir de <http://www.um.es/ead/red/M2/lopez27.pdf>

Madera Payeta, A., Monasterio Martín, I., Jaraiz Lara, A., Cantador Gutiérrez, R., Sánchez Sánchez, J. C., & Varas Moreno, R. (2013). Estudio de casos. Recuperado a partir de [http://www.uam.es/personal\\_pdi/stmaria/jmurillo/InvestigacionEE/Presentaciones/Curso\\_10/EstCasos\\_Trabajo.pdf](http://www.uam.es/personal_pdi/stmaria/jmurillo/InvestigacionEE/Presentaciones/Curso_10/EstCasos_Trabajo.pdf)

Maldonado J, Siguencia J, & Carvallo J. (2015, junio). Repositorios Educativos: Estudio de la Situación Actual y Estrategia para Mejorar su Uso Efectivo en las Universidades Ecuatorianas. *VAEP-RITA*, 3(2). Recuperado a partir de <http://rita.det.uvigo.es/VAEPRITA/201506/uploads/VAEP-RITA.2015.V3.N2.A5.pdf>

Marzal García-Quismondo, M. Á., Calzada Prado, J., & Cuevas Cerveró, A. (2006, Diciembre). DESARROLLO DE UN ESQUEMA DE METADATOS PARA LA DESCRIPCIÓN DE RECURSOS EDUCATIVOS: EL PERFIL DE APLICACIÓN MIMETA DEVELOPMENT OF A METADATA SCHEME TO DESCRIBE EDUCATIONAL RESOURCES: THE MIMETA APPLICATION PROFILE, (REVISTA ESPAÑOLA DE DOCUMENTACIÓN CIENTÍFICA), 21.

Menéndez D., V. H., Castellanos B., M. E., & Zapata G., A. (2012, julio 10). La plataforma AGORA. Una propuesta para la gestión e intercambio de objetos de aprendizaje. Recuperado a partir de [http://bioinfo.uib.es/~joemiro/aenui/procJenui/Jen2012/p279hu\\_lapl.pdf](http://bioinfo.uib.es/~joemiro/aenui/procJenui/Jen2012/p279hu_lapl.pdf)

- Metadata Innovation Dublin Core. (2012, julio). Dublin Core Metadata Element Set, Version 1.1. Recuperado a partir de <http://dublincore.org/documents/dces/>
- Miguel Rebollo Pedruelo. (2004, Diciembre). *El estándar SCORM para EaD*. Recuperado a partir de <http://www.mrebollo.es/pubs/tesina.pdf>
- Murua Olalde, J., & Urbano, F. G. (2013, de Enero de). Aplicación de las wikis semánticas como herramienta de gestión del conocimiento operativo en un departamento de TICs. Recuperado a partir de <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/18976/3/jmuruaTFC0113memoria.pdf>
- OpenDOAR. (2016, diciembre 7). Usage of Open Access Repository Software - Worldwide. Recuperado a partir de <https://goo.gl/nvRhB>
- Ouafia, G., Hélène Abel, M., & Moulin, C. (2008, julio 1). Learning Object Indexing Tool Based on a LOM Ontology. Recuperado a partir de <http://ieeexplore.ieee.org/document/4561771/authors>
- Red Iris. (2011). Dublin Core en castellano. Recuperado a partir de <http://www.rediris.es/search/dces/>
- Rincón Valadez, M. I., Martínez Lazcano, V., & Curiel Anaya, A. (2012). Evaluación con Objetos de Aprendizaje en el Ambiente ELearning MOODLE Mediante la Integración de Módulos Multimedia y el Instrumento HEODAR. Recuperado a partir de <http://lacro.org/papers/index.php/lacro/article/viewFile/65/60>
- Rodriguez, J., & Sulé, A. (2008). DSpace: un manual específico para gestores de la información y la documentación. Recuperado a partir de <http://bid.ub.edu/20rodri2.htm>
- Santacruz-Valencia, L. P., Aedo, I., & Delgado Kloos, C. (2004, Enero). Objetos de aprendizaje: Tendencias dentro de la web semántica. Recuperado a partir de <https://www.rediris.es/difusion/publicaciones/boletin/66-67/ponencia18.pdf>



Temesio, S. (2015, octubre). Metadatos para recursos educativos. Recuperado a partir de  
<http://www.palabraclave.fahce.unlp.edu.ar/article/view/PCv5n1a03/6893>  
w3c. (2013, mayo 30). Apache Jena. Recuperado a partir de  
[https://www.w3.org/2001/sw/wiki/Apache\\_Jena](https://www.w3.org/2001/sw/wiki/Apache_Jena)